



TUGAS AKHIR - TM141585

**PERANCANGAN SISTEM DETEKSI DAN PELACAKAN
MULTI OBJEK YANG BERGERAK DENGAN METODE
OPTICAL FLOW DAN PYRAMIDAL LUCAS-KANADE**

**HAFIZH NAZHAR PAHLEVI
NRP. 0211134000053**

**DOSEN PEMBIMBING
ARIF WAHJUDI, S.T., M.T., Ph.D.**

**DEPARTEMEN TEKNIK MESIN
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2018**

[Halaman ini sengaja dikosongkan]



FINAL PROJECT - TM141585

**DESIGN OF DETECTION AND MOVING MULTI
OBJECT TRACKING SYSTEM USING OPTICAL FLOW
AND PYRAMIDAL LUCAS-KANADE METHODE**

**HAFIZH NAZHAR PAHLEVI
NRP. 02111340000053**

**ADVISOR
ARIF WAHJUDI, S.T., M.T., Ph.D.**

**MECHANICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute Of Technology
Surabaya 2018**

[Halaman ini sengaja dikosongkan]

**PERANCANGAN SISTEM DETEKSI DAN PELACAKAN
MULTI OBJEK YANG BERGERAK DENGAN METODE
OPTICAL FLOW DAN PYRAMIDAL LUCAS-KANADE**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Teknik
Pada
Program studi S-1 Departemen Teknik Mesin
Fakultas Teknik Industri
Institut Teknologi Sepuluh Nopember

Oleh :

HAFZIH NAZHAR PAHLEVI

NRP. 02111340000053

Disetujui oleh tim penguji tugas akhir :

1. Arif Wahjudi, S.T., M.Sc., Ph.D.

NIP. 195811061986011002

2. Dinny Harnany, S.T., M.Sc.

NIP. 2100201405001

3. Ari Kurniawan Saputra, S.T., M.T.

NIP. 198604012015041001



SURABAYA
JULI, 2018

[Halaman ini sengaja dikosongkan]

PERANCANGAN SISTEM DETEKSI DAN PELACAKAN MULTI OBJEK YANG BERGERAK DENGAN METODE OPTICAL FLOW DAN PYRAMIDAL LUCAS-KANADE

Nama : Hafizh Nazhar Pahlevi
NRP : 02111340000053
Departemen : Teknik Mesin FTI-ITS
Dosen Pembimbing : Arif Wahjudi, S.T., M.T., Ph.D.

ABSTRAK

Dewasa ini, perkembangan militer khususnya terkait persenjataan sangat pesat. Hal ini dapat dilihat dari mekanisme alat senjata sebagai eksekutor utama dalam penentuan berbagai target yang dituju. Target adalah suatu tujuan (output) untuk dicapai oleh penarget dengan suatu input yang melalui suatu proses tertentu. Pengukuran adalah serangkaian kegiatan yang bertujuan untuk menentukan nilai suatu besaran dalam bentuk angka (kuantitatif). Jadi, mengukur adalah suatu proses mengaitkan angka secara empirik dan objektif pada sifat-sifat objek atau kejadian nyata sehingga angka yang diperoleh tersebut dapat memberikan gambaran yang jelas mengenai objek atau kejadian yang diukur. Salah satu metode dalam pengukuran Teknik adalah dengan metode pengukuran visual suatu objek yang ditangkap oleh kamera (image processing).

Langkah-langkah dalam penelitian ini secara garis besar terdiri dari tiga langkah, yaitu proses persiapan, proses perancangan program, dan analisa data. Proses persiapan terdiri dari studi pustaka untuk mempelajari dasar-dasar dari penelitian, kemudian identifikasi dan perumusan masalah dilakukan pada penelitan. Selanjutnya adalah perancangan program. Rancangan program tersebut ditunjukkan untuk mendapatkan nilai koordinat

(x,y) , kecepatan (x',y') , percepatan (x'',y'') , sudut rotasi (θ_x) , dan sudut elevasi (θ_y) objek pada layar.

Hasil yang diperoleh dari Tugas Akhir ini adalah sebuah program yang dapat mendeteksi multi objek bergerak dalam bentuk koordinat (x,y) , kecepatan (x',y') percepatan (x'',y'') , sudut rotasi (θ_x) , dan sudut elevasi (θ_y) pada layar. Verifikasi objek dilakukan pada mobil mainan dalam kondisi intensitas cahaya tinggi (kondisi terang) dengan tingkat frames per second (fps) kamera maksimal sebesar 30 fps. Tingkat rata-rata eror program pada verifikasi yaitu, kecepatan sumbu x sebesar 2.72%, kecepatan sumbu y sebesar 2.43%, percepatan sumbu x sebesar 2.04%, percepatan sumbu y sebesar 2.62%, sudut rotasi sebesar 0.64%, dan sudut elevasi sebesar 1.51%.

Kata kunci : Image processing, Optical Flow, Pyramidal Lucas-Kanade, mono camera, deteksi dan lacak objek.

DESIGN OF DETECTION AND MOVING MULTI OBJECT TRACKING SYSTEM USING OPTICAL FLOW AND PYRAMIDAL LUCAS-KANADE METHODE

Name : Hafizh Nazhar Pahlevi
NRP : 02111340000053
Department : Teknik Mesin FTI-ITS
Advisor : Arif Wahjudi, S.T., M.T., Ph.D.

ABSTRACT

Nowadays, militaristic weaponry had rapid development. This happened because mechanism of the weapon as a main executor for desired target. Target is an output desired by a subject with an input to by passing certain process. Measuring is a activity for to scalling an object. So measuring is a process to connect numbers on empirical and objective way so the numbers we got describing the object that we measure. One of the method is visual measurement nowadays we called image processing.

Steps in this project as we outlined consists of three steps, preparation, design of program and analyzing data. Preparation consists of studying literature and then identification of problems, and then designing the program. The purpose of the program is to get coordinate, velocity, acceleration, rotational and elevation angle.

From this project we can get a program that can detect a moving multi object in the form of coordinates, velocity, acceleration, rotational and elevation angle. Object verification is done in the toy car on high intensity light condition (bright condition) with maximum fps amounts 30 fps. Error average on verification program is consists of : x velocity 2.72%, y velocity 2.43%, x acceleration 2.04%, y acceleration 2.62%, rotational angle 0.64%, and elevation angle 1.51%.

Keywords : Image processing, Optical Flow, Pyramidal Lucas-Kanade, mono camera, detection and tracking object.

KATA PENGANTAR

Puji dan syukur marilah kita panjatkan kehadiran Tuhan Yang Maha Esa. Karena setiap hidayah dan karunia-Nya, penulis mampu menyelesaikan Tugas Akhir yang berjudul **Perancangan Sistem Deteksi dan Pelacakan Multi Objek yang Bergerak dengan Metode Optical Flow dan Pyramidal Lucas-Kanade** . Ucapan terima kasih penulis sampaikan kepada:

1. **Allah SWT** yang telah memberi semua anugrah, berkah, dan hidayahnya kepada penulis.
2. **Kedua orang tua** penulis yang selalu mendoakan, mendidik, dan memberi semangat putra-putrinya untuk melakukan yang terbaik.
3. **Kakak dan Adik** penulis yang telah memberikan banyak nasehat dan dukungan terbaiknya kepada penulis.
4. **Bapak Arif Wahjudi, S.T., M.T., Ph.D.**, selaku dosen pembimbing Tugas Akhir yang selalu memberikan ilmu, kritik dan saran, serta bimbingannya selama proses penyelesaian Tugas Akhir ini.
5. **Ibu Dinny Harnany, S.T., M.Sc.**, dan **Bapak Ari Kurniawan Saputra, S.T, M.T.** selaku dosen penguji Tugas Akhir, yang telah menyempatkan berbagi ilmu, sehingga tugas akhir ini dapat terselesaikan dengan baik.
6. **Nela** selaku Guru, **Mas'ud Asadullah** dan **Ikhsan Abi Nubli** selaku teman seperjuangan dalam pembuatan Tugas Akhir ini serta kawan-kawan P3 dan kawan-kawan Jenderal Besar (hormat syek).

Surabaya, Juli 2018

Hafizh Nazhar Pahlevi

[Halaman ini sengaja dikosongkan]

DAFTAR ISI

JUDUL.....	I
LEMBAR PENGESAHAN.....	V
ABSTRAK.....	VII
KATA PENGANTAR.....	XI
DAFTAR ISI.....	XIII
DAFTAR GAMBAR.....	XV
DAFTAR TABEL.....	XVII
BAB I.....	1
PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	2
1.2 Rumusan Masalah	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II.....	5
TINJAUAN PUSTAKA	5
2.1 Tinjauan Pustaka	5
2.2 Dasar Teori	6
2.2.1 Optical Flow	6
2.2.2 Algoritma Lucas-Kanade.....	7
2.2.3 Pyramidal Lucas-Kanade.....	8
2.2.4 Pelacakan Objek (<i>Object Tracking</i>)	9
2.2.5 Gambar Digital	9
2.2.6 Kedalaman Gambar	10
2.2.7 Sistem Koordinat	10
2.2.8 Kecepatan dan Percepatan	11
BAB III.....	13
METODE PENELITIAN	13
3.1 <i>Flowchart</i> Metodologi Penelitian.....	13
3.2 Persiapan.....	14
3.3 Perancangan Program.....	14

3.3.1 Cara Kerja.....	15
3.3.2 Pengerjaan Program.....	16
3.3.3 <i>Flowchart</i> Perancangan Program	18
3.4 Tahap Implementasi	19
BAB IV	21
PERANCANGAN DAN PEMBUATAN PROGRAM	21
4.1 Implementasi Program dalam Sistem	21
4.2 Konstruksi Program.....	22
4.2.1 Program Pendeteksian dan Pelacakan Objek.....	22
4.2.2 Program Kalibrasi.....	31
BAB V.....	35
PENGUJIAN PROGRAM.....	35
5.1 Pendeteksian dan Pelacakan Objek	35
5.1.1 Proses Pengujian Kondisi Terang.....	36
5.1.2 Proses Pengujian Kondisi Gelap.....	39
5.2 Verifikasi Pendeteksian dan Pelacakan Objek	42
5.2.1 Verifikasi Nilai Kecepatan	43
5.2.2 Verifikasi Nilai Percepatan.....	44
5.2.3 Verifikasi Nilai Sudut.....	45
5.2.4 Verifikasi Objek dan Spesifikasi Program	46
BAB VI	49
KESIMPULAN DAN SARAN	49
6.1 Kesimpulan.....	49
6.2 Saran.....	49
DAFTAR PUSTAKA	51
LAMPIRAN.....	53
Lampiran 1. Program Deteksi dan Lacak Objek	53
Lampiran 2. Tabel Verifikasi Objek.....	61
BIODATA PENULIS	63

DAFTAR GAMBAR

Gambar 2.1 Pola aliran pada optical flow	6
Gambar 2.2 Pyramidal optical flow	8
Gambar 2.3 Sistem koordinat 3 dimensi	10
Gambar 3.1 Diagram alir metodologi penelitian	13
Gambar 3.2 Posisi kamera pada alat pelontar peluru	15
Gambar 3.3 Tampilan layar dalam program.....	17
Gambar 3.4 Diagram alir pemograman	18
Gambar 4.1 Diagram alir proses deteksi koordinat objek	23
Gambar 4.2 Diagram alir proses deteksi kecepatan objek	25
Gambar 4.3 Diagram alir proses deteksi percepatan objek	26
Gambar 4.4 Diagram alir proses deteksi sudut objek.....	28
Gambar 4.5 Titik tengah pada sumbu x dan sumbu y	29
Gambar 4.6 Gambar layar pendeteksian dan pelacakan objek....	31
Gambar 4.7 Gambar hasil pendeteksian dan pelacakan objek	31
Gambar 4.8 Diagram alir proses kalibrasi kamera	32
Gambar 4.9 Papan catur untuk proses kalibrasi.	33
Gambar 4.10 Hasil dari proses kalibrasi kamera.....	33
Gambar 5.1 Objek untuk pengujian	35
Gambar 5.2 Sketsa jarak kamera dengan objek.....	36
Gambar 5.3 Hasil pendeteksian objek mobil mainan terang.....	36
Gambar 5.4 Nilai keluaran pengujian mobil mainan terang.....	37
Gambar 5.5 Hasil pendeteksian objek botol kondisi terang	37
Gambar 5.6 Nilai keluaran pengujian botol kondisi terang.....	38
Gambar 5.7 Hasil pendeteksian sisi lingkaran tabung terang.....	38
Gambar 5.8 Nilai keluaran sisi lingkaran tabung terang	39
Gambar 5.9 Nilai keluaran pengujian mobil mainan gelap.....	39
Gambar 5.10 Hasil pendeteksian objek mobil mainan gelap	40
Gambar 5.11 Hasil pendeteksian objek botol kondisi gelap	40
Gambar 5.12 Nilai keluaran pengujian botol kondisi gelap	41

Gambar 5.13 Nilai keluaran sisi lingkaran tabung gelap.....	41
Gambar 5.14 Hasil pendeteksian sisi lingkaran tabung gelap	42

DAFTAR TABEL

Tabel 5.1	Hasil verifikasi data pengujian.....	48
Tabel 5.2	Spesifikasi program.....	49

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, perkembangan militer khususnya terkait persenjataan sangat pesat. Hal ini dapat dilihat dari mekanisme alat senjata sebagai eksekutor utama dalam penentuan berbagai target yang dituju. Target adalah suatu tujuan (output) untuk dicapai oleh penarget dengan suatu input yang melalui suatu proses tertentu. Pengukuran adalah serangkaian kegiatan yang bertujuan untuk menentukan nilai suatu besaran dalam bentuk angka (kuantitatif). Jadi, mengukur adalah suatu proses mengaitkan angka secara empirik dan objektif pada sifat-sifat objek atau kejadian nyata sehingga angka yang diperoleh tersebut dapat memberikan gambaran yang jelas mengenai objek atau kejadian yang diukur. Salah satu metode dalam pengukuran Teknik adalah dengan metode pengukuran visual suatu objek yang ditangkap oleh kamera (*image processing*).

Image processing merupakan metode pengolahan data berupa gambar untuk mendapatkan gambar yang lebih disempurnakan atau untuk memperoleh informasi yang dibutuhkan. Dalam pengukuran, *image processing* dapat digunakan untuk memperoleh data berupa jarak maupun posisi. *Input* yang berupa gambar atau video dapat diperoleh dengan menggunakan baik mono kamera maupun stereo kamera. *Image processing* sendiri terbagi menjadi tiga tahap, yaitu *pre-processing*, *processing*, dan *post-processing*. *Pre-processing* sendiri adalah proses *thresholding* dimana gambar disegmentasi agar dapat membedakan objek yang akan diproses dengan latar belakangnya. Kemudian *processing* merupakan tahap pendeteksian objek pada gambar. Tahap terakhir adalah *post-processing* dimana pengguna mengekstraksi informasi yang dibutuhkan dari gambar dengan. Gambar yang diperoleh dari kamera nantinya dapat diolah dengan menggunakan bantuan perangkat lunak, contohnya *Visual Studio*,

untuk memperoleh informasi yang dibutuhkan, misalnya jarak suatu objek dari kamera.

Salah satu penerapan *image processing* dalam persenjataan militer terdapat pada alat pelontar peluru otomatis. Saat ini sudah ada alat pelontar peluru yang dapat menembak target secara otomatis dengan mengetahui koordinat dari target tersebut. Alat ini memanfaatkan *image processing* untuk mendeteksi dan mendapatkan data jarak serta koordinat 2 dimensi, kecepatan, dan percepatan objek terhadap kamera. Dengan menggunakan 2 kamera (stereo kamera), alat ini dapat mengetahui jarak dan koordinat x dan y dari target. Hanya saja aplikasi yang sudah diterapkan ini memiliki jangkauan yang terbatas oleh sudut 2 kamera yang dihasilkan (*blind area*) dan objek yang dideteksi masih terbatas oleh pengaruh cahaya dan. Oleh sebab itu, 2 kamera yang diterapkan belum bisa membidik (*tracking*) suatu objek yang bergerak dengan optimal [1].

Berdasarkan permasalahan diatas maka sistem pelacakan dan penentuan koordinat 2 dimensi, kecepatan, dan percepatan objek pada kamera tunggal akan digunakan pada penelitian ini. Sebuah aplikasi pendeteksian suatu target objek diusulkan dengan memanfaatkan metode *image processing* dan perangkat kamera. Aplikasi ini nantinya akan digunakan pada alat pelontar peluru *auto-tracking*.

1.2 Rumusan Masalah

Dari uraian pada latar belakang dapat dirumuskan masalah sebagai berikut:

1. Bagaimana menerapkan metode Optical Flow dan Pyramidal Lucas-Kanade dalam penentuan dan pelacakan multi objek yang bergerak?
2. Bagaimana menentukan nilai koordinat 2 dimensi (x, y), kecepatan (x', y'), percepatan (x'', y'') sudut rotasi (θ_x), dan sudut elevasi (θ_y) objek pada layar?

1.3 Batasan Masalah

Untuk menyederhanakan tugas akhir ini, maka rumusan masalah yang akan dibahas antara lain:

1. Menggunakan kamera tunggal (*mono camera*).
2. Jarak maksimum pelacakan adalah tiga meter.

1.4 Tujuan Penelitian

Tujuan dari proposal tugas akhir ini adalah sebagai berikut:

1. Dapat menerapkan metode Optical Flow dan Pyramidal Lucas-Kanade dalam penentuan dan pelacakan multi objek yang bergerak.
2. Dapat menentukan nilai koordinat 2 dimensi (x,y) , kecepatan (x',y') , percepatan (x'',y'') sudut rotasi (θ_x) , dan sudut elevasi (θ_y) objek pada layar.

1.5 Manfaat Penelitian

Manfaat dari proposal tugas akhir ini adalah sebagai berikut:

1. Dapat digunakan sebagai alat ukur jarak.
2. Dapat dijadikan sebagai referensi dalam bidang *image processing* dalam penelitian berikutnya.
3. Dapat digunakan untuk pengembangan otomatisasi mesin-mesin militer.

[Halaman ini sengaja dikosongkan]

BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Dalam beberapa tahun belakangan ini, teknologi di bidang pengukuran terutama dalam hal pendeteksian objek memiliki pertumbuhan yang amat pesat. Teknologi yang diterapkan pun semakin canggih seiring kebutuhan manusia untuk bertahan hidup. Metode *image processing* dengan kamera dapat digunakan untuk pendeteksian objek. Metode tersebut sudah diterapkan dalam berbagai penelitian.

Pada penelitian terdahulu, metode *triangulation* digunakan untuk penentuan jarak suatu objek terhadap kamera. Perangkat lunak *visual studio* dengan bahasa C yang diperoleh dari *library* *openCV* digunakan pada penelitian ini. Parameter yang digunakan dalam penelitian ini adalah nilai *focal length* dari kamera. Nilai *focal length* sendiri dapat diperoleh dari program kalibrasi yang telah disediakan oleh *openCV*. Kekurangan dari penelitian ini adalah daerah yang tidak tertangkap oleh kedua kamera (*blind spot*) terlalu luas sehingga tidak dapat mendeteksi objek jarak dekat [2].

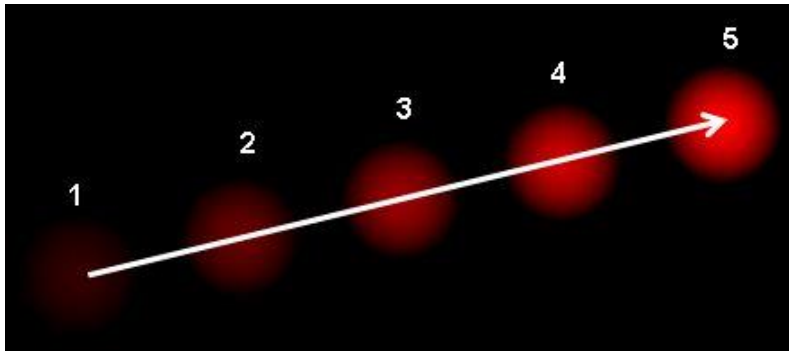
Penggunaan metode pendeteksian acuan *range* warna objek pada setiap *frame* digunakan pada penelitian selanjutnya. Sampel sebanyak 1000 *frame* dengan kecepatan 50 *fps* (*frame per second*) dibutuhkan pada metode ini. Ukuran gambar yang digunakan pada penelitian ini sebesar 352 x 288 piksel. Objek yang digunakan pada penelitian ini adalah bola dan pendeteksiannya dilakukan dengan pendekatan histogram warna RGB (*Red Green Blue*). Namun, jarak objek dengan kamera tidak dapat ditentukan pada penelitian ini karena hanya menggunakan mono kamera [3].

Penentuan jarak sumber cahaya terhadap kamera dilakukan pada penelitian selanjutnya dan dapat diukur dengan metode perbandingan jumlah piksel. Dalam penelitian ini, sumber cahaya berasal dari laser berwarna hijau yang kemudian ditangkap oleh kamera. Hasil gambar yang ditangkap oleh kamera nanti

diubah menjadi gambar biner, Dari gambar biner, banyaknya piksel berwarna putih akan dihitung. Setelah itu, banyaknya nilai piksel dari tiap gambar akan diproses selanjutnya dalam penelitian ini [4].

2.2 Dasar Teori

2.2.1 Optical Flow



Gambar 2.1 Pola aliran pada optical flow [5]

Optical Flow adalah perkiraan gerakan suatu bagian dari sebuah citra berdasarkan turunan intensitas cahayanya pada sebuah sekuen citra. Dalam koordinat 2 dimensi dinyatakan sebagai seberapa jauh suatu piksel citra berpindah diantara dua frame citra yang berurutan, tetapi dalam koordinat 3 dimensi dinyatakan sebagai seberapa jauh suatu volume piksel (*voxel*) berpindah pada dua volume yang berurutan. Perhitungan turunan dilakukan berdasarkan perubahan intensitas cahaya pada kedua frame citra maupun volume. Perubahan intensitas cahaya pada suatu bagian citra dapat disebabkan oleh gerakan yang dilakukan oleh objek, gerakan sumber cahaya, ataupun perubahan sudut pandang [4]. Pola aliran optical flow dalam setiap perubahan *frame* berurutan dapat dilihat pada gambar 2.1. Persamaan optical flow adalah sebagai berikut [5] :

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (1)$$

dengan pendekatan taylor series dan membagi dt maka persamaan menjadi :

$$f_x u + f_y v + f_t = 0 \quad (2)$$

dimana,

$$f_x = \partial f / \partial x \quad (3)$$

$$f_y = \partial f / \partial y \quad (4)$$

$$u = dx / dt \quad (5)$$

$$v = dy / dt \quad (6)$$

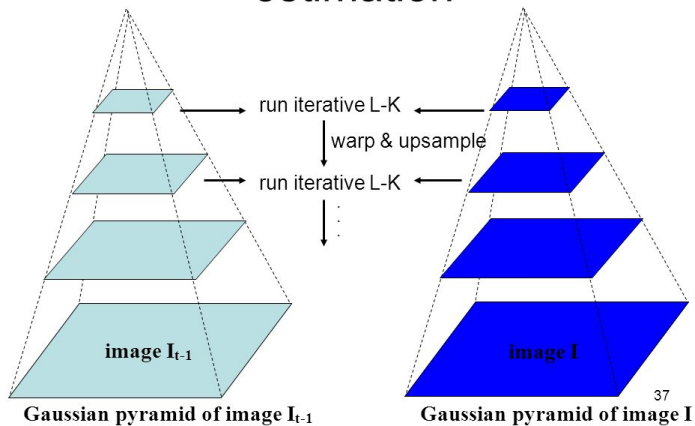
2.2.2 Algoritma Lucas-Kanade

Algoritma Lucas-Kanade, pertama kali diajukan pada tahun 1981, awalnya adalah sebuah usaha untuk mencari teknik registrasi citra yang cepat dengan memanfaatkan *gradient* intensitas spasial. Pada perkembangannya, algoritma ini kemudian menjadi salah satu algoritma optical flow yang penting. Berbeda dengan algoritma Horn-Schunk yang bekerja berbasis pada keseluruhan citra, algoritma ini bekerja berdasar pada informasi *local* yang diturunkan dari window kecil (*patch*) disekeliling titik yang diperhitungkan. Kelemahan digunakan window local kecil pada algoritma Lucas-Kanade adalah tidak terdeteksinya gerakan-gerakan yang besar karena gerakan-gerakan tersebut jatuh diluar window. Permasalahan ini kemudian dapat diatasi dengan mengimplementasikan penyelesaian dengan prinsip piramida, yaitu Pyramidal Lucas-Kanade. Prinsip ini merupakan

penyelesaian berdasar iterasi dari level detail citra paling rendah hingga level detail citra paling tinggi [6].

2.2.3 Pyramidal Lucas-Kanade

Coarse-to-fine optical flow estimation



Gambar 2.2 Pyramidal optical flow [6]

Penyelesaian algoritma Lucas-Kanade dengan pendekatan piramida, atau disebut Pyramidal Lucas-Kanade diajukan pertama kali oleh Jean-Yves Bouguet. Pendekatan ini menggunakan prinsip piramida, yaitu bekerja dari detail citra paling rendah hingga detail citra paling tinggi. Tujuannya adalah agar gerakan yang “besar” dapat diperhitungkan. Sementara asumsi yang digunakan pada algoritma Lucas-Kanade adalah gerakan yang “kecil” dan koheren, sehingga tidak dapat menangkap gerakan yang “besar”. Solusi untuk dapat menangkap gerakan yang “besar” pada algoritma Lucas-Kanade adalah dengan menggunakan window yang besar. Tetapi penggunaan window yang besar sering kali membuat gerakan yang ditangkap adalah gerakan yang tidak koheren.

Algoritma Pyramidal Lucas-Kanade menyelesaikan permasalahan tersebut tanpa menghilangkan asumsi gerakan yang koheren.

Algoritma Pyramidal Lucas-Kanade pertama bekerja pada *layer* piramida paling tinggi. Kemudian hasilnya digunakan sebagai titik awal untuk bekerja pada *layer* dibawahnya. Hal ini berlanjut hingga mencapai level paling rendah [6]. Skema Pyramidal optical flow dapat dilihat pada gambar 2.2.

2.2.4 Pelacakan Objek (*Object Tracking*)

Pelacakan objek pada citra merupakan cara mengikuti suatu elemen citra yang telah dilokalisasi menjadi objek bergerak dalam tahapan waktu secara otomatis dalalam gambar / video bergerak. Pelacakan memegang peranana penting dalam pengolahan video karena dapat menjadi tolok ukur untuk mengekstraksi beberapa perkiraan properti objek yang bergerak dalam video [7].

2.2.5 Gambar Digital

Gambar merupakan representasi objek fisik 3 dimensi kedalam bentuk 2 dimensi. Bentuk dari gambar sendiri bermacam-macam, ada yang berupa gambar berwarna, ada yang hanya berupa gambar hitam-putih. Kumpulan gambar yang ditampilkan dengan kecepatan *frame per second* tertentu akan membentuk suatu video. Gambar digital merupakan gambar dua dimensi yang dapat ditampilkan pada layar komputer dalam bentuk piksel.

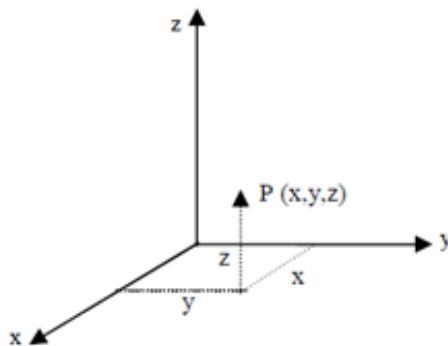
Gambar digital tersusun dari piksel-piksel berbentuk matrik dari fungsi intensitas cahaya dengan ukuran $M \times N$. Fungsi intensitas cahaya dapat ditulis sebagai $f(x,y)$ yang mana f merupakan nilai amplitudo pada koordinat x dan y . Secara matematis, gambar digital dapat dituliskan seperti persamaan (1) :

$$f(x,y) = f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \quad (7)$$

2.2.6 Kedalaman Gambar

Nilai yang biasa digunakan untuk menentukan kualitas suatu gambar adalah banyaknya piksel yang dimiliki suatu gambar. Semakin banyak atau padat piksel yang ada pada suatu gambar maka kualitas atau resolusi yang dimiliki gambar tersebut juga semakin meningkat. Piksel sendiri merupakan representasi titik terkecil dari suatu gambar yang memiliki informasi yang nantinya dapat diproses. [1].

2.2.7 Sistem Koordinat



Gambar 2.3 Sistem koordinat 3 dimensi [1]

Sistem koordinat kartesius merupakan sistem yang digunakan untuk menentukan posisi suatu titik, garis maupun gambar dalam suatu bidang baik 2 dimensi maupun 3 dimensi. Dalam sistem koordinat 2 dimensi, digunakan dua garis yang saling tegak lurus dalam satu bidang sebagai sumbu koordinat. Sumbu yang ke arah horizontal diberi label axis-x, sedangkan sumbu yang ke arah vertikal diberi label axis-y. Dalam bidang 3 dimensi, ditambahkan satu sumbu lagi yang arahnya tegak lurus terhadap bidang xy. Sumbu yang tegak lurus terhadap bidang xy ini diberi label sumbu z. Sumbu-sumbu pada koordinat kartesian

ini bersifat orthogonal (antar sumbu saling tegak lurus) [1]. Gambar sistem koordinat 3 dimensi dapat dilihat pada gambar 2.3.

2.2.8 Kecepatan dan Percepatan

Kecepatan rata-rata ditentukan berdasarkan perpindahan benda. Perpindahan benda adalah vektor penghubung posisi awal dengan posisi akhir sehingga selalu berupa garis lurus. Perpindahan benda tidak menyatakan lintasan yang ditempuh benda. Panjang lintasan yang ditempuh benda lebih besar atau sama dengan panjang perpindahan benda. Panjang lintasan dan besar perpindahan sama hanya jika benda bergerak dalam garis lurus. Kecepatan rata-rata adalah rasio antara jarak tempuh dengan waktu tempuh. Karena jarak tempuh umumnya lebih besar daripada besar perpindahan maka laju rata-rata umumnya lebih besar daripada besar kecepatan rata-rata. Laju rata-rata didefinisikan sebagai:

$$v = \frac{s}{\Delta t} \quad (8)$$

dimana, s adalah jarak tempuh dan Δt waktu tempuh.

Besaran yang mengukur berapa cepat kecepatan berubah dinamakan percepatan. Percepatan didefinisikan sebagai perbandingan antara perubahan kecepatan benda dengan lama kecepatan tersebut berubah. Misalkan saat t_1 kecepatan sesaat benda adalah v_1 dan pada saat t_2 kecepatan sesaat benda adalah v_2 , maka perubahan kecepatan benda adalah

$$\Delta v_{21} = v_2 - v_1 \quad (9)$$

Lama waktu kecepatan berubah adalah

$$\Delta t_{21} = t_2 - t_1 \quad (10)$$

Definisi percepatan adalah

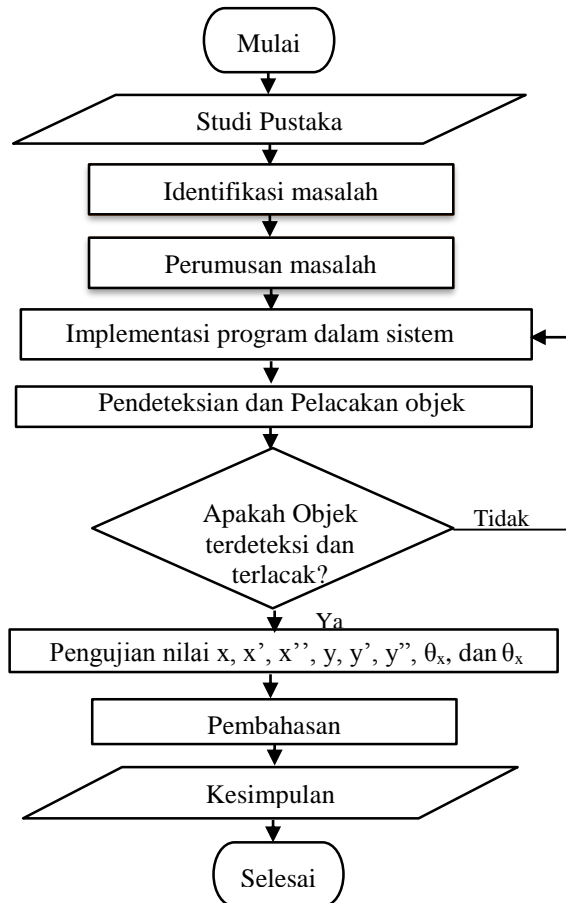
$$a = \frac{\Delta v_{z1}}{\Delta t} \quad (11)$$

Percepatan juga merupakan besaran vektor dan pada perhitungan percepatan tidak dipengaruhi nilai kecepatan pada berbagai waktu. Yang dibutuhkan pada percepatan adalah nilai kecepatan pada saat awal dan saat akhir [8].

BAB III METODE PENELITIAN

Pada Bab 3 ini akan dibahas mengenai langkah-langkah sistematis yang akan dijadikan sebagai acuan kerangka penelitian.

3.1 *Flowchart* Metodologi Penelitian



Gambar 3.1 Diagram alir metodologi penelitian

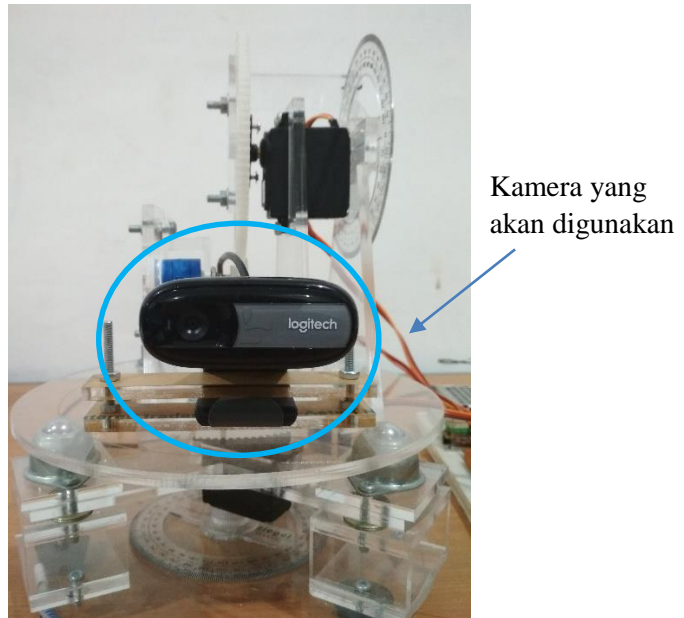
Langkah-langkah dalam penelitian ini secara garis besar terdiri dari tiga langkah, yaitu proses persiapan, proses perancangan program, dan analisa data. Proses persiapan terdiri dari studi pustaka untuk mempelajari dasar-dasar dari penelitian, kemudian identifikasi dan perumusan masalah dilakukan pada penelitian. Selanjutnya adalah perancangan program. Rancangan program tersebut ditunjukkan untuk mendapatkan nilai koordinat (x,y) , kecepatan (x',y') , percepatan (x'',y'') , sudut rotasi (θ_x) , dan sudut elevasi (θ_y) objek pada layar. Setelah mendapatkan hasil pendeteksian, tahapan memasuki proses akhir yaitu analisa untuk pembahasan hasil penelitian sehingga dapat ditarik kesimpulan dari tugas akhir ini.

3.2 Persiapan

Proses persiapan terdiri dari tiga tahapan yaitu studi pustaka, identifikasi masalah dan perumusan masalah. Studi pustaka merupakan acuan referensi untuk mendalami permasalahan yang akan diteliti. Pada penggunaan aplikasi program digunakan studi pustaka dengan memanfaatkan buku-buku referensi maupun jurnal-jurnal yang berkaitan dengan permasalahan yang akan dibahas. Setelah itu, identifikasi masalah untuk menyusun sistem pendeteksi dan penentu koordinat, kecepatan, dan percepatan objek pada layar dilakukan pada penelitian. Terakhir, tahap perumusan masalah berisi perancangan program untuk pendeteksi dan penentu koordinat, kecepatan, dan percepatan objek pada layar.

3.3 Perancangan Program

Pada proses ini, nilai x , x' , x'' , y , y' , y'' , θ_x , dan θ_y objek pada layar ditentukan pada program. Program yang dirancang pada penelitian ini akan diaplikasikan pada *prototype* alat pelontar peluru. Skema dari alat tersebut dapat dilihat pada gambar 3.2



Gambar 3.2 Posisi kamera pada alat pelontar peluru

Pada gambar 3.2 posisi kamera berada pada pergerakan sumbu absis alat pelontar peluru. Kamera diposisikan tepat ditengah dari alat agar pendeteksian dan pelacakan objek menjadi optimum dalam daya tangkap kamera dan pendeteksian objek pada layar menjadi akurat. Selain itu, skema ini membatasi kamera untuk mendeteksi dan melacak pergeseran sumbu ordinat. Setelah diperoleh variabel dari kamera, data hasil akan ditampilkan pada program yang terpasang diperangkat.

3.3.1 Cara Kerja

Cara kerja dari program adalah pertama kamera dikalibrasi untuk mengetahui nilai intrinsik pada kamera. Selanjutnya adalah objek dalam bentuk gambar 2 dimensi dideteksi pada program. Kemudian data gambar dianalisa dan diproses menggunakan

program C++. Setelah itu, proses pendeteksian koordinat (x,y) pada layar dilakukan dalam program. Dengan nilai waktu pandeteksian serta perubahan nilai koordinat, dalam program diperoleh kecepatan dari objek (x',y') . Setelah mendapatkan nilai kecepatan, perubahan kecepatan ini akan digunakan untuk mencari nilai percepatan objek (x'',y'') . Penentuan sudut rotasi (θ_x) , dan sudut elevasi (θ_y) merupakan proses akhir yang dilakukan.

3.3.2 Pengerjaan Program

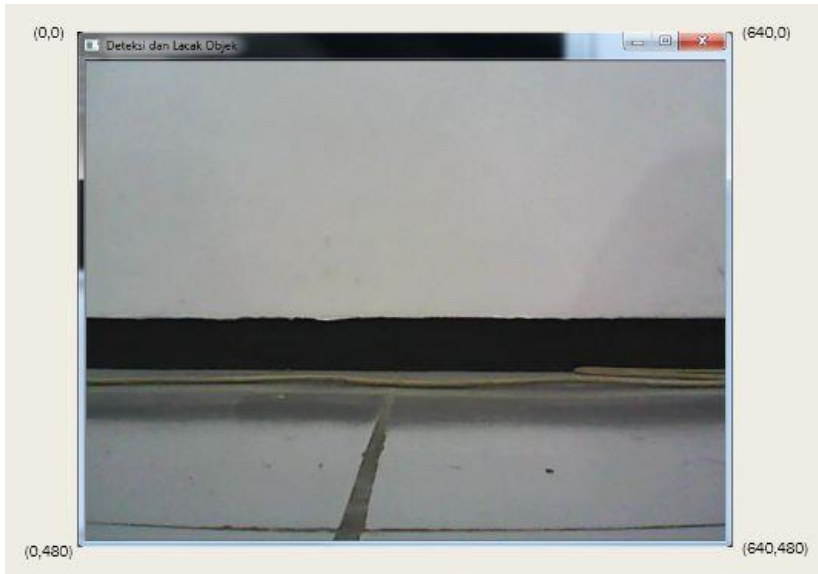
Terdapat dua tahapan pengerjaan perancangan program, yaitu kalibrasi kamera dan penentuan koordinat (x,y) , kecepatan (x',y') , percepatan (x'',y'') , sudut rotasi (θ_x) , dan sudut elevasi (θ_y) objek pada layar.

1. Kalibrasi kamera

Program kalibrasi kamera dibuat untuk mengetahui nilai intrinsik pada kamera. Data – data yang diperoleh yaitu *focal length* (f), *imaging 2D coordinate system* (u, v), titik gambar 2D kamera kanan dan kiri, dan matriks kamera kanan dan kiri (opsional).

2. Penentuan koordinat (x,y) , kecepatan (x',y') , percepatan (x'',y'') , sudut rotasi (θ_x) , dan sudut elevasi (θ_y) objek pada layar

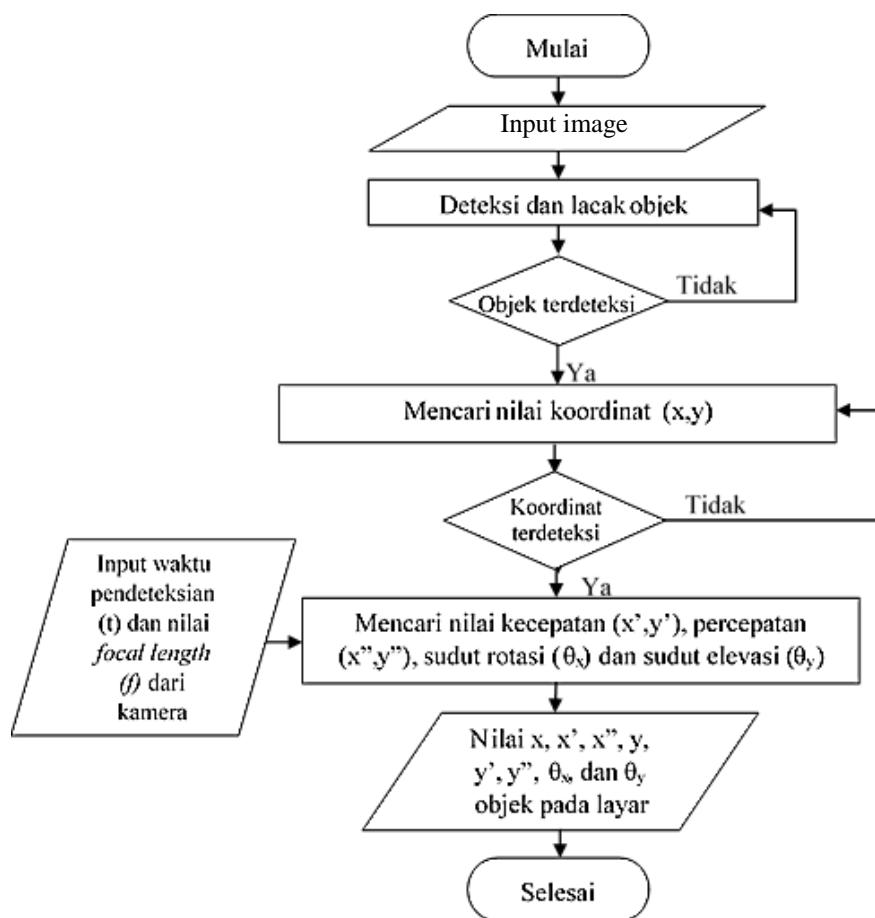
Penentuan koordinat x dan y pada layar berupa titik (*point*) saat target sudah dikunci. Nilai dari koordinat x dan y dibatasi oleh resolusi pixel (640, 480) dengan susunan seperti pada gambar dibawah ini.



Gambar 3.3 Tampilan layar dalam program

Setelah program mendeteksi koordinat x dan y , waktu pergerakan objek (t) akan digunakan dan disubstitusikan ke persamaan (8) dan diperoleh perubahan pergerakan atau kecepatan objek yang dinyatakan dengan x' dan y' . Pada tahap akhir, perubahan kecepatan objek yang dideteksi dan waktu pergerakan objek (t) akan disubstitusikan ke persamaan (10). Lalu, percepatan objek yang dinyatakan dalam x'' dan y'' akan diperoleh dalam program. Selanjutnya nilai sudut rotasi (θ_x) dan sudut elevasi (θ_y) akan ditentukan dari nilai *focal length* (f).

3.3.3 Flowchart Perancangan Program



Gambar 3.4 Diagram alir pemograman

Pada diagram alir yang ditunjukkan pada gambar 3.5, dapat dilihat bahwa tahapan yang pertama kali adalah *image processing*. Tahap selanjutnya, deteksi dan lacak objek dilakukan dengan metode Optical Flow dan Pyramidal Lucas-Kanade. Setelah objek terdeteksi, nilai koordinat (x,y) diperoleh pada program. Dengan input waktu pendeteksian (t) dari pelacakan objek, nilai kecepatan (x',y') dan percepatan (x'',y'') diperoleh pada program. Dengan input *focal length* (f) , nilai sudut rotasi (θ_x) dan sudut elevasi (θ_y) akan diperoleh pada program. Setelah itu, nilai yang diperoleh pada program akan dinyatakan pada layar.

3.4 Tahap Implementasi

Tahap implementasi penelitian ini merupakan tahap pengimplementasian *software* yang telah dirancang pada alat pelontar peluru. Tahap implementasi diawali dengan menjalankan program dalam berbagai variasi posisi objek. Kemudian dari hasil deteksi objek tersebut diperoleh data berupa koordinat (x,y) , kecepatan (x',y') , percepatan (x'',y'') , sudut rotasi (θ_x) , dan sudut elevasi (θ_y) objek pada layar dari berbagai posisi. Data ini kemudian dijadikan bahan analisa penentuan nilai variabel posisi, kecepatan, dan percepatan objek terhadap *prototype* alat pelontar peluru.

[Halaman ini sengaja dikosongkan]

BAB IV

PERANCANGAN DAN PEMBUATAN PROGRAM

4.1 Implementasi Program dalam Sistem

Program pendeteksian dan pelacakan koordinat 2 dimensi, kecepatan, percepatan, dan sudut objek pada layar disusun dengan menggunakan teknologi *computer vision*. Perangkat lunak yang digunakan dalam pembuatan program adalah Visual Studio Community 2015 dan OpenCV 3.1. Mesin pengolah yang digunakan untuk menjalankan program adalah *personal computer* dengan spesifikasi sebagai berikut:

- a. ASUS model X401U-WX076D
- b. *Processor* AMD C-60 APU with Radeon™ HD Graphics 1.00 GHz
- c. *Memory* (RAM) 2.00 GB (1.61 GB usable)
- d. *Operating System* Windows 7 Ultimate (64-bit)

Dari spesifikasi diatas, penentuan performa program yang dibangun perlu dilakukan pengujian. Pengujian terdiri dari dua jenis, yang pertama pendeteksian multi objek dalam intensitas cahaya rendah (kondisi gelap) maupun intensitas cahaya tinggi (kondisi terang) dan yang kedua adalah verifikasi pendeteksian objek tunggal. Multi objek yang akan diuji adalah mobil mainan, botol, dan sisi lingkaran tabung, sedangkan objek tunggal yang akan diverifikasi adalah botol. Pengambilan data dilakukan secara *real-time* pada siang dan malam hari. Setelah pengambilan data maka nilai verifikasi objek dapat diketahui. Selain itu, data dari spesifikasi program atau kemampuan program dalam mengambil data keluar program dapat dicari. Data spesifikasi program dinyatakan dalam nilai maksimum data dalam program.

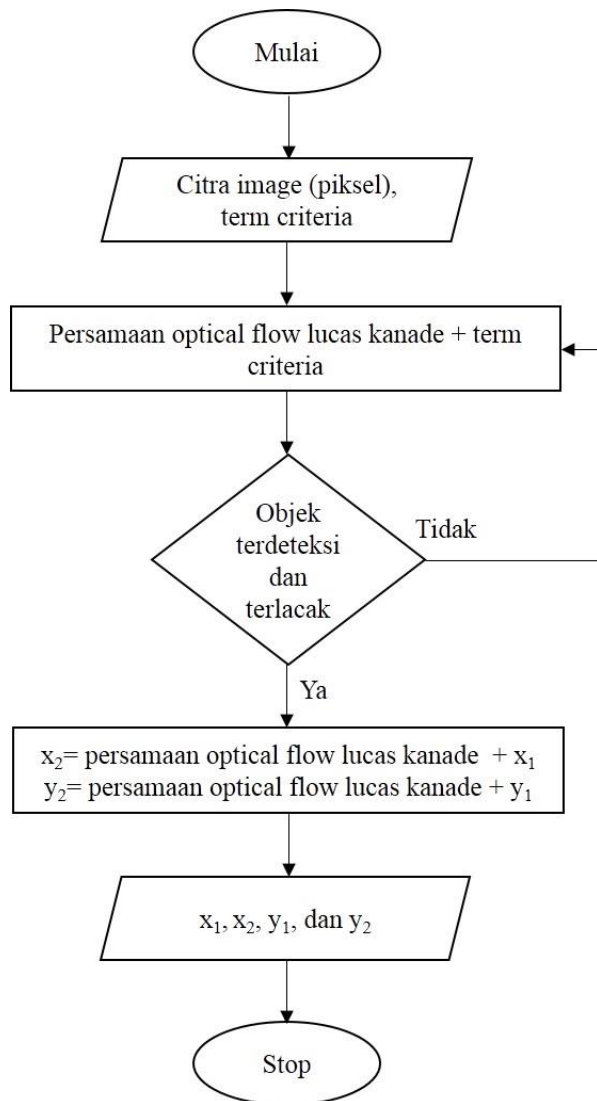
4.2 Konstruksi Program

Pada tahap konstruksi program dibahas mengenai tahap-tahap penerapan metode yang digunakan dalam pembuatan program pendeteksian dan pelacakan koordinat 2 dimensi (x,y) , kecepatan (x',y') , percepatan (x'',y'') , serta posisi sudut (θ_x, θ_y) pada layar. Proses konstruksi program dimulai dari tahap pendeteksian objek. Tahap selanjutnya adalah menentukan koordinat 2 dimensi objek (x,y) . Setelah itu menentukan kecepatan objek (x',y') . Tahap selanjutnya adalah penentuan percepatan (x'',y'') dan tahap terakhir adalah menentukan posisi sudut (θ_x, θ_y) . Konstruksi program dibangun ke dalam kode program menggunakan perangkat yang telah ditentukan.

4.2.1 Program Pendeteksian dan Pelacakan Objek

Kode program 1 : Deteksi koordinat

```
point = Point2f((float)x, (float)y);
TermCriteria termcrit(TermCriteria::COUNT |
TermCriteria::EPS, 20, 0.03);
calcOpticalFlowPyrLK(prevGray, gray,
points[0], points[1], status, err, winSize, 3,
termcrit, 0, 0.001); size_t i, k;
simpan_point1[1] = points[0]; points[1][k++]
= points[1][i]; circle(image, points[1][i], 3,
Scalar(0, 255, 0), -5, 8);
int x_awal = simpan_point1[1][0].x; int
y_awal = simpan_point1[1][0].y;
int x_akhir = points[1][0].x; int y_akhir =
points[1][0].y;
```

Gambar 4.1 Diagram alir proses deteksi koordinat objek

Pada tahap pendeteksian dan pelacakan objek, proses pertama yang dilakukan deteksi koordinat. Tahapan pertama dalam deteksi koordinat adalah menentukan citra objek yang akan terdeteksi menggunakan kamera tunggal lalu menentukan *term criteria* dari program yang akan dijalankan pada program. Setelah menentukan *term criteria* dari program, persamaan opical flow lucas kanade akan diproses untuk menguji apakah objek terdeteksi maupun terlacak atau tidak. Ketika objek tidak terdeteksi maka program akan secara otomatis kembali dalam penentuan *term criteria* maupun persamaan optical flow lucas kanade dengan mengatur parameter pada tiap persamaan.

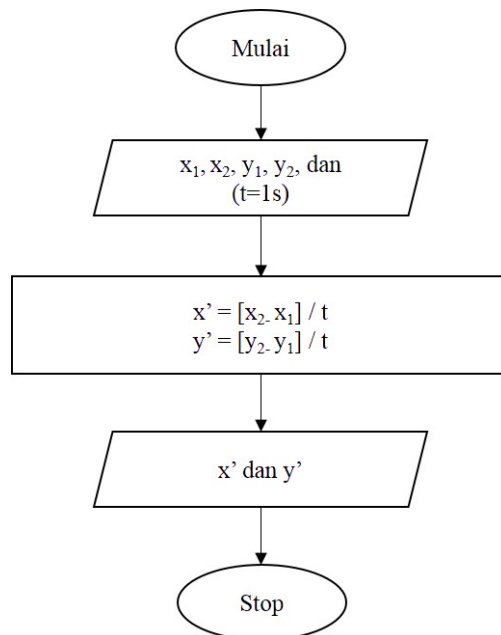
Tahap selanjutnya ketika objek sudah dapat terdeteksi maupun dilacak maka akan diperoleh posisi pada layar, posisi tersebut adalah berebentuk x dan y pada 2 sumbu geometri. Posisi x pada program akan disimpan menjadi x awal dan posisi y pada program akan disimpan menjadi y awal. Setelah diperoleh x awal dan y awal, program akan menentukan x akhir maupun y akhir. Persamaan tersebut seperti yang tertera pada *flowchart* program. Setelah diproses melalui persamaan maka program pendeteksian memiliki 4 keluaran yaitu, x awal, x akhir, y awal, dan y akhir. Diagram alir proses deteksi koordinat dapat dilihat pada gambar 4.1.

Proses kedua dari tahap pendeteksian dan pelacakan objek adalah deteksi kecepatan. Pada proses pertama diperoleh keluaran x awal, x akhir, y awal, dan y akhir. Keluaran tersebut selanjutnya diproses dengan waktu pengambilan yang di *setup* tiap 1 detik. Persamaan yang digunakan pada pendeteksian kecepatan objek tertera pada persamaan (8). Hasil yang diperoleh dari program deteksi kecepatan adalah nilai kecepatan pada sumbu x (x')

dan kecepatan pada sumbu y (y'). Diagram alir proses deteksi kecepatan dapat dilihat pada gambar 4.2.

Kode Program 2 : Deteksi kecepatan

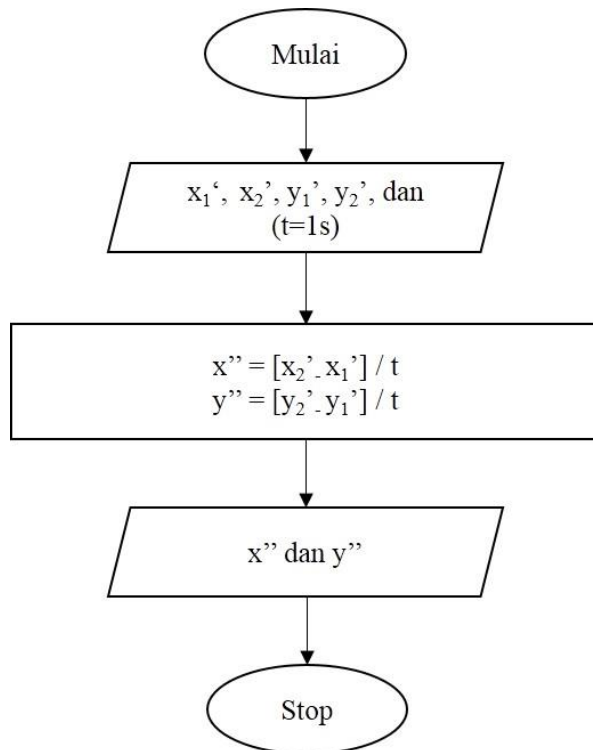
```
printf("waktu pengambilan = %.2fs\n",
(double)(CLOCKS_PER_SEC * 1) /
CLOCKS_PER_SEC);
perpindahanX = abs(x_akhir - x_awal) / 1;
kecepatanX = perpindahanX /
((double)(CLOCKS_PER_SEC * 1) /
CLOCKS_PER_SEC);
perpindahanY = abs(y_akhir - y_awal) / 1;
kecepatanY = perpindahanY /
((double)(CLOCKS_PER_SEC * 1) /
CLOCKS_PER_SEC);
```



Gambar 4.2 Diagram alir proses deteksi kecepatan objek

Kode Program 3 : Deteksi percepatan

```
printf("waktu pengambilan = %.2fs\n",
(double)(CLOCKS_PER_SEC * 1) /
CLOCKS_PER_SEC);
percepatanX = (kecepatanX - kec0X ) /
((double)(CLOCKS_PER_SEC * 1) /
CLOCKS_PER_SEC);
percepatanY = (kecepatanY - kec0Y ) /
((double)(CLOCKS_PER_SEC * 1) /
CLOCKS_PER_SEC);
```

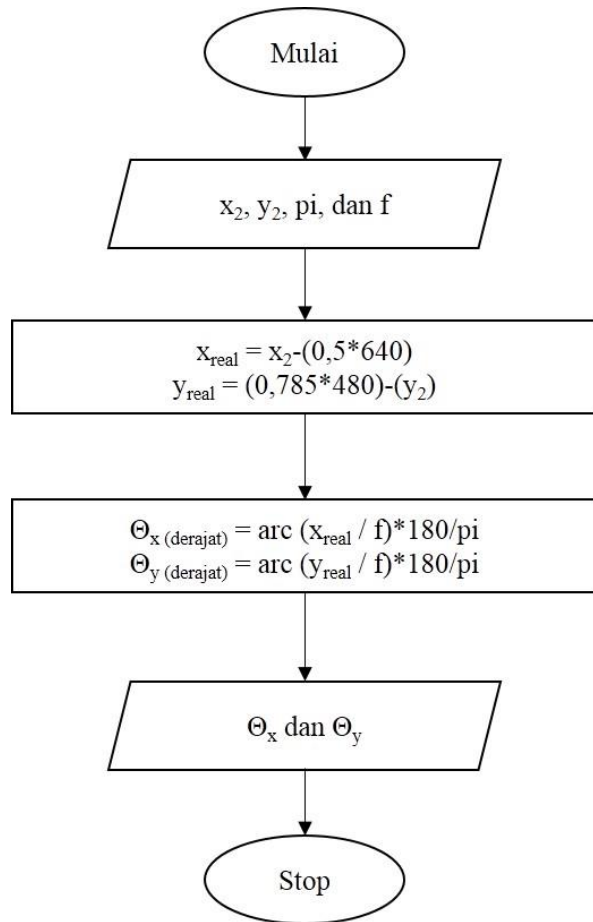


Gambar 4.3 Diagram alir proses deteksi percepatan objek

Proses ketiga dari tahap pendeteksian dan pelacakan objek adalah deteksi percepatan. Pada proses kedua diperoleh keluaran kecepatan pada sumbu x (x') dan kecepatan pada sumbu y (y'). Adanya pengambilan waktu program menyebabkan program akan memperoleh kecepatan selanjutnya yang didefinisikan menjadi kecepatan akhir pada sumbu x maupun y. Kecepatan awal dan akhir tersebut akan terdefinisi pada program menjadi 4 keluaran, yaitu kecepatan awal pada sumbu x (x'), kecepatan awal pada sumbu y (y'), kecepatan akhir pada sumbu x (x'') dan kecepatan akhir pada sumbu y (y''). Keluaran tersebut selanjutnya diproses dengan waktu pengambilan yang di *setup* tiap 1 detik. Persamaan yang digunakan pada pendeteksian percepatan objek tertera pada persamaan (11). Hasil yang diperoleh dari program deteksi percepatan adalah nilai percepatan pada sumbu x (x'') dan percepatan pada sumbu y (y''). Diagram alir proses deteksi percepatan dapat dilihat pada gambar 4.3.

Kode Program 4 : Deteksi sudut

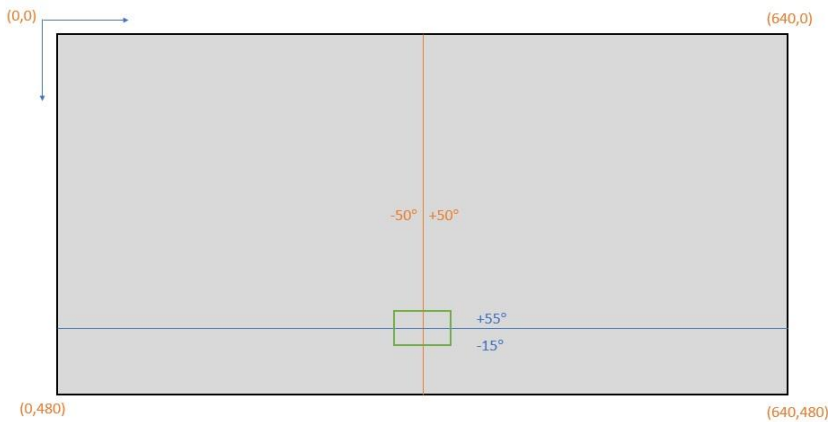
```
#define pi 3.14159265
double focal_length2 = 1969.93290799997;
double tangenX = (Xreal2 / focal_length2);
double tangenY = (Yreal2 / focal_length2);
double sudutX = Math::Atan(tangenX) *
180/pi;
double sudutY = Math::Atan(tangenY) *
180/pi
```



Gambar 4.4 Diagram alir proses deteksi sudut objek

Proses keempat dari tahap pendeteksian dan pelacakan objek adalah deteksi sudut. Pada proses deteksi sudut dibutuhkan parameter khusus dari tiap yaitu nilai *focal length* (f). Nilai *focal length* sendiri diperoleh dari program kalibrasi kamera yang merupakan program lain

yang akan dibahas nanti. Setelah tiap parameter diketahui, maka proses selanjutnya adalah menentukan titik tengah dari alat pelontar peluru pada tiap sumbu untuk menentukan nilai koordinat sebenarnya dalam program. Hal itu dilakukan agar pada tahap implementasi akan sesuai dengan kontrol penentuan target. Penentuan titik tengah dari alat pelontar peluru tersebut adalah sebagai berikut.



Gambar 4.5 Titik tengah pada sumbu x dan sumbu y

1. Sumbu X

Pada gambar 4.5 dapat diketahui bahwa sudut searah sumbu x negatif adalah 50° dan searah sumbu x positif adalah 50°. Titik tengah tepat diantara perpotongan dari 640 pixel searah sumbu x, sehingga persamaan untuk menentukan sumbu x sebenarnya dalam program dapat dilihat pada persamaan 12.

$$x_{real} = x_{akhir} - (0,5 \times 640) \quad (12)$$

2. Sumbu Y

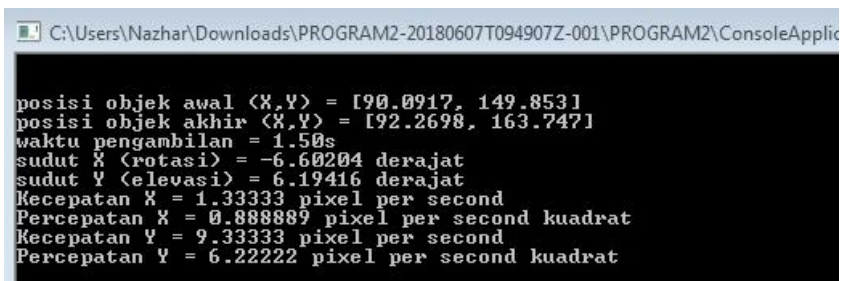
Pada gambar 4.5 dapat diketahui bahwa sudut searah sumbu y positif adalah 55° dan searah sumbu y negatif adalah 15° . Titik tengah terdapat diantara 11/14 bagian searah sumbu y (dalam program tertulis 0,785) dari 480 pixel, sehingga persamaan untuk menentukan sumbu y dapat dilihat pada persamaan 13.

$$y_{real} = (0,785 \times 480) - y_{akhir} \quad (13)$$

Setelah menentukan sumbu koordinat sebenarnya dalam program, tahap selanjutnya adalah memproses persamaan penentuan sudut dalam derajat yang sudah tertera pada program. Hasil yang diperoleh dari persamaan tersebut adalah nilai sudut pada sumbu x (sudut rotasi, θ_x) dan nilai sudut pada sumbu y (sudut elevasi, θ_y). Diagram alir proses deteksi sudut dapat dilihat pada gambar 4.4. Berikut adalah tampilan dari tampilan layar dan hasil keluaran pada program dapat dilihat pada gambar 4.6 dan 4.7.



Gambar 4.6 Gambar layar pendeteksian dan pelacakan objek

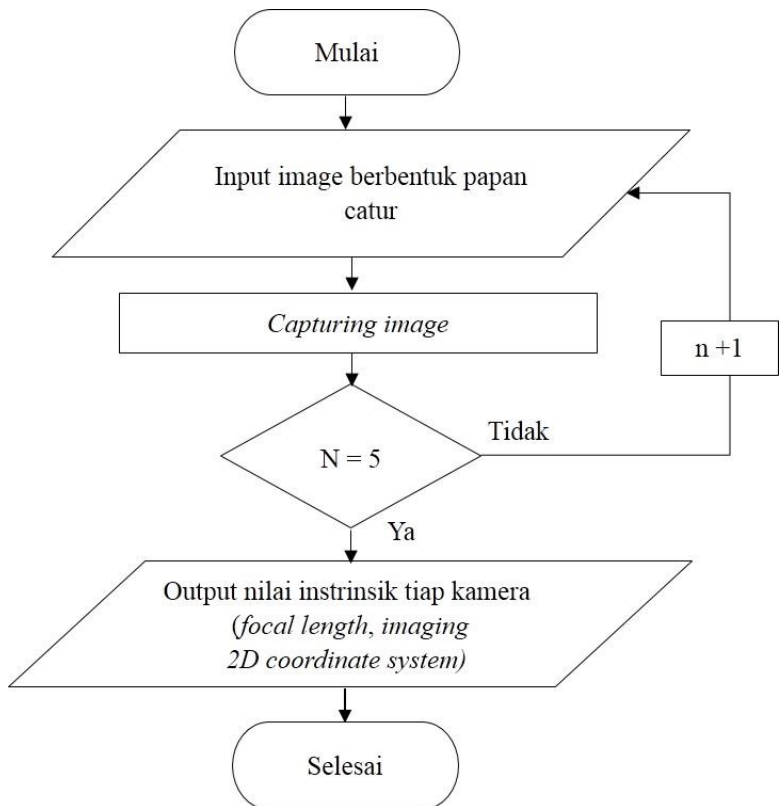


Gambar 4.7 Gambar hasil pendeteksian dan pelacakan objek

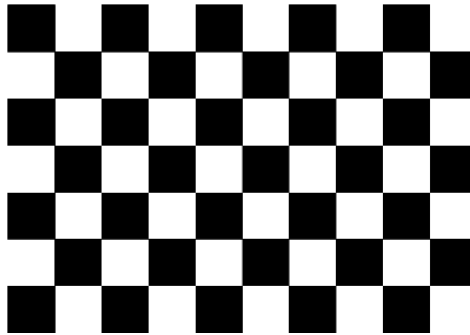
4.2.2 Program Kalibrasi

Proses kalibrasi dilakukan terlebih dahulu sebelum pengujian dilakukan. Program kalibrasi kamera sendiri telah disediakan oleh OpenCV namun perlu diberi penyesuaian dengan data yang digunakan. Data yang

digunakan dalam proses kalibrasi adalah gambar kotak hitam dan putih seperti papan catur dengan ukuran 10 kotak x 7 kotak. Ukuran tiap kotaknya adalah 2,5 cm x 2,5 cm. Gambar papan catur yang digunakan dapat dilihat pada gambar 4.8, sedangkan untuk diagram alir dari proses kalibrasi dapat dilihat pada gambar 4.9.



Gambar 4.8 Diagram alir proses kalibrasi kamera



Gambar 4.9 Papan catur untuk proses kalibrasi

```
%YAML:1.0
calibration_time: "Fri May 25 15:18:23 2018"
image_width: 640
image_height: 480
board_width: 9
board_height: 6
square_size: 1.
aspectRatio: 1.
flags: 2
camera_matrix: !!opencv-matrix
  rows: 3
  cols: 3
  dt: d
  data: [ 1.9699329079999757e+03, 0., 1.2184463473242494e+02, 0.,
    1.9699329079999757e+03, 2.4261819365914965e+02, 0., 0., 1. ]
distortion_coefficients: !!opencv-matrix
  rows: 5
  cols: 1
  dt: d
  data: [ 2.1635526075649378e-01, 1.4672920559445778e+01,
    5.5911334271912813e-03, 5.3462308177970118e-02,
    -2.3748183694097463e+02 ]
```

Gambar 4.10 Hasil dari proses kalibrasi kamera

Hasil dari proses kalibrasi kamera dapat dilihat pada gambar 4.10. Dalam proses kalibrasi kamera diperoleh nilai *focal length* (f) dan sistem koordinat 2D kamera. Hasil kalibrasi diperoleh dalam bentuk matriks seperti berikut:

$$M = \begin{bmatrix} 1.9699329079999757e + 03 & 0 & 1.2184463473242494e + 02 \\ 0 & 1.9699329079999757e + 03 & 2.4261819365914965e + 02 \\ 0 & 0 & 1 \end{bmatrix}$$

Nilai dari f , u_0 , dan v_0 yang diperoleh dari proses kalibrasi dapat diketahui berdasarkan bentuk matriks M sebagai berikut:

$$M = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Berdasarkan bentuk matriks M tersebut dapat diketahui bahwa nilai f dari kamera adalah $1.9699329079999757e+03$ yang selanjutnya akan digunakan untuk mencari nilai sudut rotasi (θ_x) dan sudut elevasi (θ_y).

BAB V

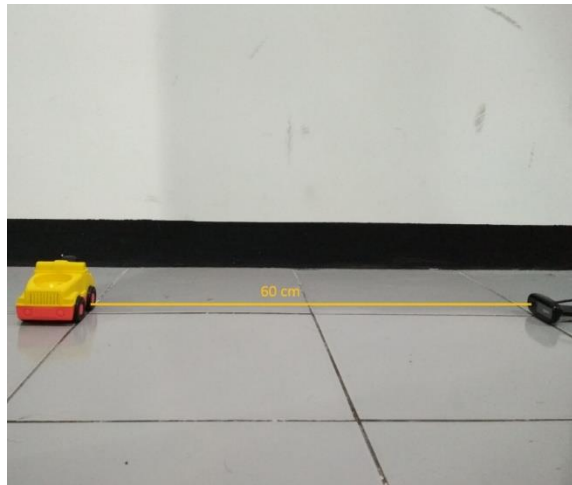
PENGUJIAN PROGRAM

5.1 Pendeteksian dan Pelacakan Objek

Pengujian deteksi dan pelacakan objek dilakukan dengan kondisi yang berbeda yaitu pada saat intensitas cahaya tinggi (kondisi terang) dan intensitas cahaya rendah (kondisi gelap). Objek yang digunakan dalam pengujian ada 3 jenis yaitu, mobil mainan, botol, dan sisi lingkaran tabung. Objek pengujian dapat dilihat pada gambar 5.1 dan sketsa jarak pada objek dapat dilihat pada gambar 5.2. Jarak antara kamera dan objek yang diunakan adalah sejauh 60 cm. Pada penentuan nilai kecepatan maupun percepatan digunakan waktu pengambilan tiap pengujian sebesar 1,5 detik sedangkan untuk proses verifikasi pengambilan data tiap 1 detik. Untuk Nilai keluaran akan ditampilkan dalam layar dan juga *tab windows* pada program.



Gambar 5.1 Objek untuk pengujian



Gambar 5.2 Sketsa jarak kamera dengan objek

5.1.1 Proses Pengujian Kondisi Terang



Gambar 5.3 Hasil pendeteksian objek mobil mainan terang

```

C:\Users\Nazhar\Downloads\PROGRAM2-20180607T094907Z-001\PROGRAM2\ConsoleAp
posisi objek awal (X,Y) = [539.323, 382.113]
posisi objek akhir (X,Y) = [484.128, 383.484]
waktu pengambilan = 1.50s
sudut X (rotasi) = 4.75899 derajat
sudut Y (elevasi) = -0.180327 derajat
Kecepatan X = 36.6667 pixel per second
Percepatan X = 24.4444 pixel per second kuadrat
Kecepatan Y = 0.666667 pixel per second
Percepatan Y = 0.444444 pixel per second kuadrat

```

Gambar 5.4 Nilai keluaran pengujian mobil mainan terang

Proses pengujian dalam kondisi terang diperoleh hasil pada 3 objek. Objek pertama adalah mobil mainan. Hasil pada pendeteksian dan pelacakan objek dapat dilihat pada gambar 5.3 dan nilai keluaran dapat dilihat pada gambar 5.4. Pengujian selanjutnya dilakukan pada botol. Hasil pada pendeteksian dan pelacakan objek dapat dilihat pada gambar 5.5 dan nilai keluaran dapat dilihat pada gambar 5.6.



Gambar 5.5 Hasil pendeteksian objek botol kondisi terang

```

C:\Users\Nazhar\Downloads\PROGRAM2-20180607T094907Z-001\PROGRAM2\ConsoleApp
posisi objek awal (X,Y) = [129.191, 334.821]
posisi objek akhir (X,Y) = [128.846, 352.079]
waktu pengambilan = 1.50s
sudut X (rotasi) = -5.56676 derajat
sudut Y (elevasi) = 0.721273 derajat
Kecepatan X = 0.666667 pixel per second
Percepatan X = 0.444444 pixel per second kuadrat
Kecepatan Y = 12 pixel per second
Percepatan Y = 8 pixel per second kuadrat

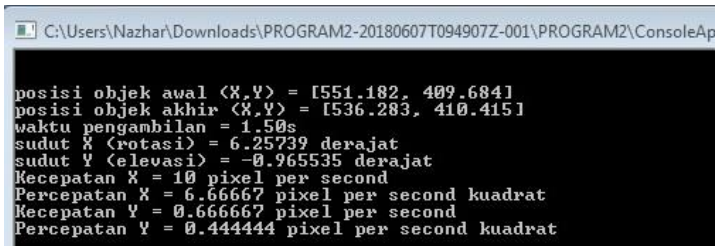
```

Gambar 5.6 Nilai keluaran pengujian botol kondisi terang

Pengujian selanjutnya dilakukan pada sisi lingkaran tabung. Hasil pada pendeteksian dan pelacakan objek dapat dilihat pada gambar 5.7 dan nilai keluaran dapat dilihat pada gambar 5.8.



Gambar 5.7 Hasil pendeteksian sisi lingkaran tabung terang



```

C:\Users\Nazhar\Downloads\PROGRAM2-20180607T094907Z-001\PROGRAM2\ConsoleAp
posisi objek awal <X,Y> = [551.182, 409.684]
posisi objek akhir <X,Y> = [536.283, 410.415]
waktu pengambilan = 1.50s
sudut X (rotasi) = 6.25739 derajat
sudut Y (elevasi) = -0.965535 derajat
Kecepatan X = 10 pixel per second
Percepatan X = 6.66667 pixel per second kuadrat
Kecepatan Y = 0.666667 pixel per second
Percepatan Y = 0.444444 pixel per second kuadrat

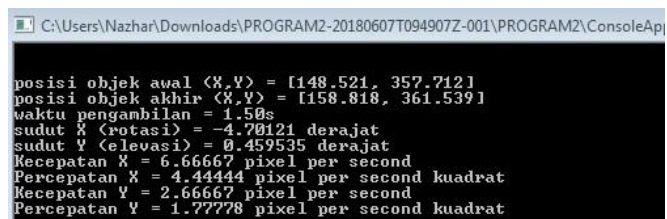
```

Gambar 5.8 Nilai keluaran sisi lingkaran tabung terang

Berdasarkan data yang diperoleh pada pengujian multi objek dengan intensitas cahaya tinggi (kondisi terang) menunjukkan bahwa tiap objek dapat terdeteksi dan terlacak dengan baik oleh program. Nilai keluaran tiap perubahan *frame* juga diperoleh, yang selanjutnya dapat menjadi input dalam kontrol alat pelontar peluru otomatis.

5.1.2 Proses Pengujian Kondisi Gelap

Proses pengujian dalam kondisi gelap diperoleh hasil pada 3 objek. Objek pertama adalah mobil mainan. Nilai keluaran dapat dilihat pada gambar 5.9 dan hasil pada pendeteksian dan pelacakan objek dapat dilihat pada gambar 5.10.

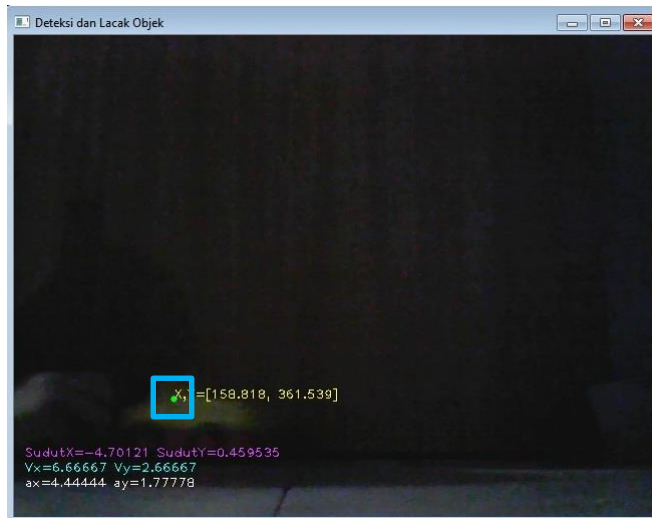


```

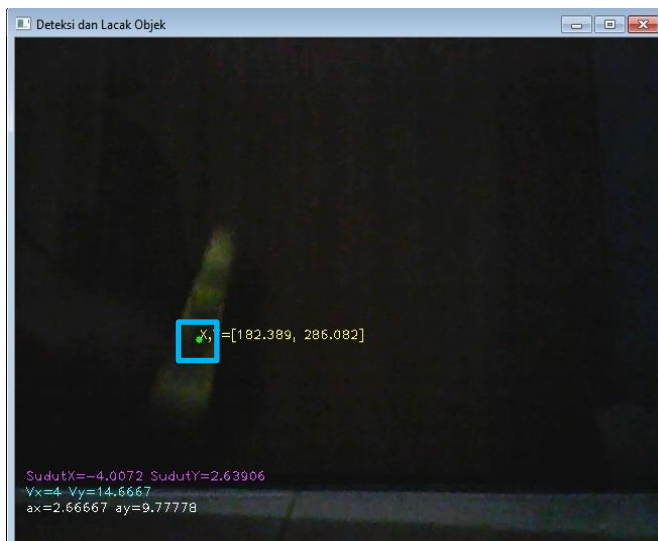
C:\Users\Nazhar\Downloads\PROGRAM2-20180607T094907Z-001\PROGRAM2\ConsoleAp
posisi objek awal <X,Y> = [148.521, 357.712]
posisi objek akhir <X,Y> = [158.818, 361.539]
waktu pengambilan = 1.50s
sudut X (rotasi) = -4.70121 derajat
sudut Y (elevasi) = 0.459535 derajat
Kecepatan X = 6.66667 pixel per second
Percepatan X = 4.44444 pixel per second kuadrat
Kecepatan Y = 2.66667 pixel per second
Percepatan Y = 1.77778 pixel per second kuadrat

```

Gambar 5.9 Nilai keluaran pengujian mobil mainan gelap

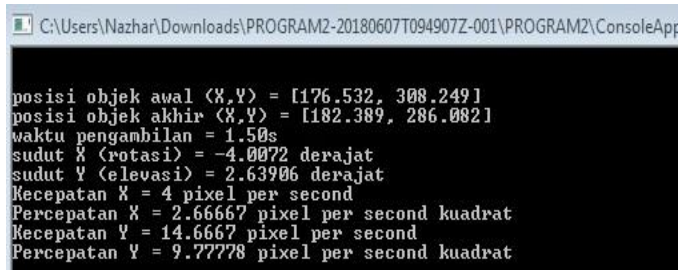


Gambar 5.10 Hasil pendeteksian objek mobil mainan gelap



Gambar 5.11 Hasil pendeteksian objek botol kondisi gelap

Pengujian selanjutnya dilakukan pada botol. Hasil pada pendeteksian dan pelacakan objek dapat dilihat pada gambar 5.11 dan nilai keluaran dapat dilihat pada gambar 5.12.

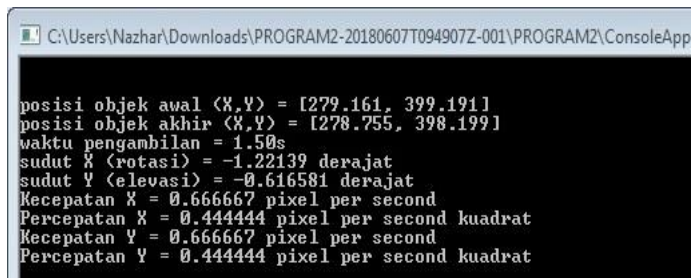


```

C:\Users\Nazhar\Downloads\PROGRAM2-20180607T094907Z-001\PROGRAM2\ConsoleApp
posisi objek awal (X,Y) = [176.532, 308.249]
posisi objek akhir (X,Y) = [182.389, 286.082]
waktu pengambilan = 1.50s
sudut X (rotasi) = -4.0072 derajat
sudut Y (elevasi) = 2.63906 derajat
Kecepatan X = 4 pixel per second
Percepatan X = 2.66667 pixel per second kuadrat
Kecepatan Y = 14.6667 pixel per second
Percepatan Y = 9.77778 pixel per second kuadrat
  
```

Gambar 5.12 Nilai keluaran pengujian botol kondisi gelap

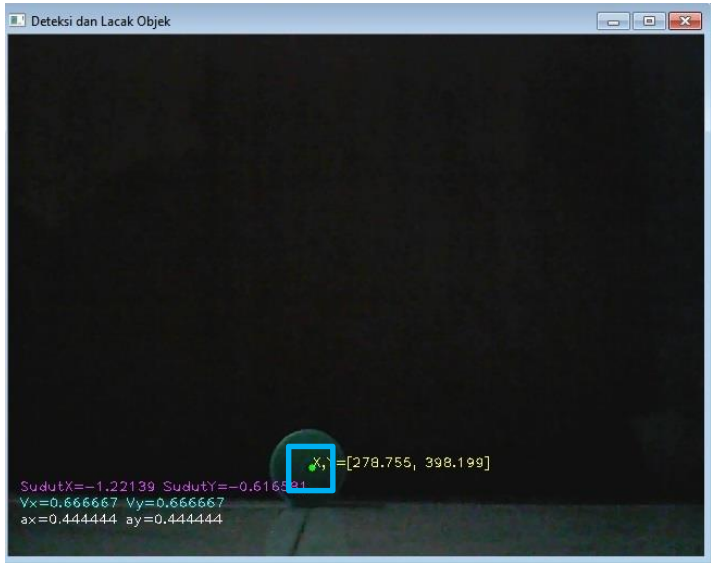
Pengujian selanjutnya dilakukan pada sisi lingkaran tabung. Nilai keluaran dapat dilihat pada gambar 5.14 dan hasil pada pendeteksian dan pelacakan objek dapat dilihat pada gambar 5.14.



```

C:\Users\Nazhar\Downloads\PROGRAM2-20180607T094907Z-001\PROGRAM2\ConsoleApp
posisi objek awal (X,Y) = [279.161, 399.191]
posisi objek akhir (X,Y) = [278.755, 398.199]
waktu pengambilan = 1.50s
sudut X (rotasi) = -1.22139 derajat
sudut Y (elevasi) = -0.616581 derajat
Kecepatan X = 0.666667 pixel per second
Percepatan X = 0.444444 pixel per second kuadrat
Kecepatan Y = 0.666667 pixel per second
Percepatan Y = 0.444444 pixel per second kuadrat
  
```

Gambar 5.13 Nilai keluaran sisi lingkaran tabung gelap



Gambar 5.14 Hasil pendeteksian sisi lingkaran tabung gelap

Berdasarkan data yang diperoleh pada pengujian multi objek dalam intensitas cahaya rendah (kondisi gelap) menunjukkan bahwa tiap objek dapat terdeteksi dan terlacak dengan baik oleh program. Nilai keluaran tiap perubahan *frame* juga diperoleh, yang selanjutnya dapat menjadi input dalam kontrol alat pelontar peluru otomatis.

5.2 Verifikasi Pendeteksian dan Pelacakan Objek

Verifikasi dilakukan untuk menentukan apakah nilai keluaran program sesuai dengan kondisi nilai sebenarnya. Nilai koordinat objek pada layar akan menjadi input pada proses verifikasi. Verifikasi pengujian dilakukan pada nilai kecepatan, percepatan dan sudut pada objek. Contoh verifikasi nilai dilakukan

pada pendeteksian dan pelacakan objek botol dalam intensitas intensitas cahaya rendah (kondisi gelap). Data keluaran yang akan diverifikasi dapat dilihat pada gambar 5.11.

5.2.1 Verifikasi Nilai Kecepatan

Verifikasi nilai kecepatan objek adalah sebagai berikut.

Diketahui :

- Posisi objek awal (x_1) = 176.532 pixel
- Posisi objek awal (y_1) = 308.249 pixel
- Posisi objek akhir (x_2) = 182.389 pixel
- Posisi objek akhir (y_2) = 286.082 pixel
- Selisih waktu (t) = 1,5 detik

Analisa :

-Kecepatan x

$$x' = \left[\frac{x_2 - x_1}{t} \right]$$

$$x' = \left[\frac{182,389 - 176,532 \text{ (pixel)}}{1,5 \text{ (detik)}} \right]$$

$$x' = 3,904 \text{ pixel/detik}$$

-Kecepatan y

$$y' = \left[\frac{y_2 - y_1}{t} \right]$$

$$y' = \left[\frac{286,082 - 308,249 \text{ (pixel)}}{1,5 \text{ (detik)}} \right]$$

$$y' = 14,778 \text{ pixel/detik}$$

Berdasarkan hasil verifikasi diperoleh nilai kecepatan pada sumbu x adalah 3,904 pixel/detik dan nilai kecepatan pada sumbu y adalah 14,778 pixel/detik, sedangkan pada program nilai kecepatan pada sumbu x adalah 4 pixel/detik dan nilai kecepatan pada sumbu y adalah 14,666 pixel/detik.

5.2.2 Verifikasi Nilai Percepatan

Verifikasi nilai percepatan objek adalah sebagai berikut.

Diketahui :

-Kecepatan objek awal (x_1') = 3,904667 pixel/detik

-Kecepatan objek akhir (y_1') = 14,778 pixel/detik

-Kecepatan objek akhir (x_2') = 8 pixel/detik

-Kecepatan objek akhir (y_2') = 28 pixel/detik

-Selisih waktu (t) = 1,5 detik

Analisa :

-Percepatan x

$$x'' = \frac{x_2' - x_1'}{t}$$

$$x'' = \frac{8 - 3,904667 \text{ (pixel/detik)}}{1,5 \text{ (detik)}}$$

$$x'' = 2,73 \text{ pixel/detik}^2$$

-Percepatan y

$$y'' = \frac{y_2' - y_1'}{t}$$

$$y'' = \frac{28 - 14,778 \text{ (pixel/detik)}}{1,5 \text{ (detik)}}$$

$$y'' = 8,814 \text{ pixel/detik}^2$$

Berdasarkan hasil verifikasi diperoleh nilai percepatan pada sumbu x adalah 2,73 pixel/detik² dan nilai percepatan pada sumbu y adalah 8,814 pixel/detik², sedangkan pada program nilai percepatan pada sumbu x adalah 2,666 pixel/detik² dan nilai percepatan pada sumbu y adalah 9,777 pixel/detik².

5.2.3 Verifikasi Nilai Sudut

Verifikasi nilai sudut objek adalah sebagai berikut.

Diketahui :

-focal length (f) = 1,9699329079999757e+03

- $\pi = 3,14$

Analisa :

-Sudut rotasi (θ_x)

$$x_{real} = x_{akhir} - (0,5 \times 640)$$

$$x_{real} = 182,389 - (0,5 \times 640)$$

$$x_{real} = -41.40139262$$

$$\theta_x = \tan^{-1}(x_{real} / f) \times (180/\pi)$$

$$\theta_x = \tan^{-1}\left(\frac{-41.40139262}{1.969}\right) \times (180/3,14)$$

$$\theta_x = -3,995^\circ$$

-Sudut elevasi (θ_y)

$$y_{real} = (0,785 \times 480) - y_{akhir}$$

$$y_{real} = (0,785 \times 480) - 286,082$$

$$y_{real} = 27.39640217$$

$$\theta_y = \tan^{-1}(y_{real}/f) \times (180/\pi)$$

$$\theta_y = \tan^{-1}\left(\frac{27.39640217}{1.969}\right) \times (180/3,14)$$

$$\theta_y = 2,646^\circ$$

Berdasarkan hasil verifikasi diperoleh nilai sudut pada sumbu x adalah $3,995^\circ$ dan nilai sudut pada sumbu y adalah, $2,646^\circ$, sedangkan pada program nilai sudut pada sumbu x adalah $-4,007^\circ$ dan nilai kecepatan pada sumbu y adalah $2,639^\circ$.

5.2.4 Verifikasi Objek dan Spesifikasi Program

Verifikasi objek dilakukan dengan pengambilan sebanyak 10 data *frame* pada program tiap 1 parameter. Sehingga total *frame* yang diverifikasi sebanyak 60 *frame*. Kamera yang digunakan memiliki tingkat *frames per second* (fps) maksimal sebesar 30 fps. Range waktu data tiap pengambilan adalah 1 detik. Dimulai dari detik ke-0 hingga detik ke-10. Tiap parameter pengambilan memiliki rata-rata error. Verifikasi dilakukan pada objek mobil

mainan dengan intensitas cahaya tinggi (kondisi terang). Data selengkapnya dapat dilihat pada lampiran.

Tabel 5.1 Hasil verifikasi data pengujian

No.	Pengujian	Sumbu	Rata-rata Error
1	Kecepatan	x	2.72%
2	Kecepatan	y	2.43%
3	Percepatan	x	2.04%
4	Percepatan	y	2.62%
5	Sudut	x	0.64%
6	Sudut	y	1.51%

Tabel 5.1 adalah hasil verifikasi data pengujian objek. Berdasarkan tabel, kesalahan terbesar terjadi pada kecepatan sumbu x dan kesalahan terkecil terjadi pada sudut x. Kesalahan verifikasi terjadi karena kemampuan dari kamera yang memiliki tingkat fps rendah sehingga transfer data tiap *frame* ke layar memiliki estimasi waktu yang lebih lama. Kesalahan juga terjadi karena kemampuan *Central Processing Unit* (CPU) pada saat menjalankan program memiliki resiko tinggi terdistraksi oleh aktivitas (program) lain yang sedang dijalankan. Pada pengujian terjadi kesalahan operator dikarenakan pada saat pengujian objek bergerak cepat searah sumbu menyebabkan kamera tidak dapat optimal mengambil tiap *frame* untuk diproses ke dalam CPU.

Tabel 5.2 Spesifikasi program

No.	Parameter	Maksimum Pendeteksian
1	Koordinat X	640 pixel
2	Koordinat Y	480 pixel
3	Sudut X	9,23 derajat
4	Sudut Y	10,83 derajat
5	Kecepatan X	24,112 pixel/detik (t = 1 detik)
6	Kecepatan Y	12,056 pixel/detik (t = 1 detik)
7	Percepatan X	18,084 pixel/detik (t = 1 detik)
8	Percepatan Y	9,042 pixel/detik (t = 1 detik)

Setelah dilakukan verifikasi objek, selanjutnya menentukan spesifikasi program. Spesifikasi program dilakukan untuk menentukan kemampuan program dalam mengirim data dari kamera kedalam CPU. Data spesifikasi program adalah nilai maksimum yang dapat dikeluarkan dalam program. Nilai maksimum dari koordinat dan sudut berdasarkan pada luasan maksimum *frame* dari layar kamera dalam program. Nilai maksimum pada kecepatan dan percepatan berdasarkan pada kemampuan maksimum waktu pengambilan data keluaran program sampai program terdeteksi. Spesifikasi program dapat dilihat pada tabel 5.2.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Beberapa kesimpulan yang diperoleh pada penelitian tugas akhir ini adalah sebagai berikut:

1. Pendeteksian dan pelacakan multi objek dengan menerapkan metode Optical Flow dan Pyramidal Lucas-Kanade menggunakan objek mobil mainan, botol, dan sisi lingkaran tabung dapat dilakukan dengan kondisi intensitas cahaya tinggi (kondisi terang) dan intensitas cahaya rendah (kondisi gelap).
2. Penentuan nilai koordinat 2 dimensi (x, y), kecepatan (x', y'), percepatan (x'', y'') sudut rotasi (θ_x), dan sudut elevasi (θ_y) objek pada layar dapat ditentukan. Dengan menggunakan koordinat, verifikasi nilai kecepatan, percepatan, dan sudut dapat dilakukan. Verifikasi objek dilakukan pada mobil mainan dalam kondisi intensitas cahaya tinggi (kondisi terang) dengan tingkat *frames per second* (fps) kamera maksimal sebesar 30 fps. Tingkat rata-rata eror pada verifikasi yaitu, kecepatan sumbu x sebesar 2.72%, kecepatan sumbu y sebesar 2.43%, percepatan sumbu x sebesar 2.04%, percepatan sumbu y sebesar 2.62%, sudut rotasi sebesar 0.64%, dan sudut elevasi sebesar 1.51%.

6.2 Saran

Berdasarkan hasil pengujian program dan pembahasan pada tugas akhir ini, muncul beberapa saran untuk mengembangkan penelitian ini.

1. Pendeteksian objek dilakukan menggunakan kamera dengan tingkat *frames per second* (fps) tinggi, agar

pada saat pelacakan program dapat melacak target dengan baik.

2. Multi objek sebaiknya definitif agar program dapat mengenai target yang sesuai.

DAFTAR PUSTAKA

- [1] Pasa, F.P. (2017). **Penentuan Koordinat 3 Dimensi Target Tunggal Pada Sistem Pelontar Peluru Autotracking Menggunakan Stereo Kamera Bersudut Beta 15 Derajat**. Surabaya. Institut Teknologi Sepuluh Nopember.
- [2] Lutfi, W.C. (2015). **Kalibrasi Stereo Kamera dan Penentuan Koordinat 3 Dimensi Untuk Target Tunggal pada Sistem Pelontar Peluru Autotracking**. Surabaya. Institut Teknologi Sepuluh Nopember.
- [3] Imron, M.A. (2015). **Pendeteksian dan Pelacakan 2D Multi Objek Secara Real – time dengan Menggunakan Kamera Tunggal**. Surabaya. Institut Teknologi Sepuluh Nopember.
- [4] Shukla, N., Trivedi, A. (2015). *Image Based Distance Measurement Technique for Robot Vision using New LBPA approach*.
- [5] **OpenCV Optical Flow**. (2017). Dipetik pada 24 Maret 2018 , dari <https://docs.opencv.org/>.
- [6] Umar U., Soelistijorini R., Darwito H.A. (2011). **Tracking Arah Gerakan Telunjuk Jari Berbasis Webcam Menggunakan Metode Optical Flow**. Surabaya. Politeknik Elektronika Surabaya.
- [7] Sigit, R., Setiawardhana, Kartika D.R. (2016). *3D Sign Language Translator Using Optical Flow*. Surabaya. Politeknik Elektronika Surabaya.
- [8] Abdullah, M. **Fisika Dasar I**. Maret 2016. Bandung. Institut Teknologi Bandung.

[Halaman ini sengaja dikosongkan]

LAMPIRAN

Lampiran 1. Program Deteksi dan Lacak Objek

```
#include "opencv2/video/tracking.hpp"
#include "opencv2/imgproc.hpp"
#include "opencv2/videoio.hpp"
#include "opencv2/highgui.hpp"
#include <stdio.h>
#include <time.h>
#include <iostream>
#include <iomanip>
#include <ctype.h>
#include <chrono>
#include <string>
#include <string.h>

using namespace System;
using namespace cv;
using namespace std;
#define pi 3.14159265

static void help()
{
    "\tESC - quit the program\n"
    "\tc - delete all the points\n"<< endl;
}

Point2f point;
bool addRemovePt = false;

static void onMouse(int event, int x, int y, int ,void*)
{
    if (event == EVENT_LBUTTONDOWN)
```

```

        {
            point = Point2f((float)x, (float)y);
            addRemovePt = true;
        }
    }

int main(int argc, char** argv)
{
    VideoCapture cap;
    TermCriteria termcrit(TermCriteria::COUNT |
TermCriteria::EPS, 20, 0.03);
    Size subPixWinSize(10, 10), winSize(31, 31);
    cv::CommandLineParser parser(argc, argv,
"{ @input[0]}");
    string input = parser.get<string>("@input");

    if (input.size() == 1 && isdigit(input[0]))
        cap.open(input[0] - '0');
    else
        cap.open(input);

    if (!cap.isOpened())
    {
        cout << "Could not initialize capturing...\n";
        return 0;
    }

    namedWindow("Deteksi dan Lacak Objek", 1);
    setMouseCallback("Deteksi dan Lacak Objek", onMouse,
0);

    Mat gray, prevGray, image, frame, kordinat;
    vector<Point2f> points[2];
    vector<Point2f> simpan_point1[2];

    for (;;)

```



```

{
    cap >> frame;
    if (frame.empty())
        break;

    frame.copyTo(image);
    cvtColor(image, gray, COLOR_BGR2GRAY);

    if (nightMode)
        image = Scalar::all(0);

    if (needToInit)
    {
        goodFeaturesToTrack(gray, points[1],
MAX_COUNT, 0.01, 10, Mat(), 3, 0, 0.04);
        cornerSubPix(gray, points[1],
subPixWinSize, Size(-1, -1), termcrit);
        addRemovePt = false;
    }
    else if (!points[0].empty())
    {
        vector<uchar> status;
        vector<float> err;
        if (prevGray.empty())
            gray.copyTo(prevGray);

        calcOpticalFlowPyrLK(prevGray, gray,
points[0], points[1], status, err, winSize,
3, termcrit, 0, 0.001);
        size_t i, k;
        simpan_point1[1] = points[0];
    }
}

```

```

for (i = k = 0; i < points[1].size(); i++)
{
    if (addRemovePt)
    {
        if (norm(point -
points[1][i]) <= 5)
        {
            addRemovePt =
false;
            continue;
        }
    }

    if (!status[i])
        continue;

    points[1][k++] = points[1][i];

    circle(image, points[1][i], 3,
Scalar(0, 255, 0), -5, 8);

}
points[1].resize(k);

std::cout << "posisi objek awal (X,Y) = "
<< simpan_point1[1][0] << "\n";
std::cout << "posisi objek akhir (X,Y) = "
<< points[1][0] << "\n" ;

int x_awal = simpan_point1[1][0].x;
int y_awal = simpan_point1[1][0].y;
int x_akhir = points[1][0].x;
int y_akhir = points[1][0].y;

double kec0X;

```

```

double perpindahanX;
double kecepatanX;
double percepatanX;
double kec0Y;
double perpindahanY;
double kecepatanY;
double percepatanY;

kec0X = 0;
kec0Y = 0;

printf("waktu pengambilan = %.2fs\n",
(double)(CLOCKS_PER_SEC * 1) / CLOCKS_PER_SEC);
perpindahanX = abs(x_akhir - x_awal) /
1;

kecepatanX = perpindahanX /
((double)(CLOCKS_PER_SEC * 1) / CLOCKS_PER_SEC);
percepatanX = (kecepatanX - kec0X) /
((double)(CLOCKS_PER_SEC * 1) / CLOCKS_PER_SEC);
perpindahanY = abs(y_akhir - y_awal) /
1;

kecepatanY = perpindahanY /
((double)(CLOCKS_PER_SEC * 1) / CLOCKS_PER_SEC);
percepatanY = (kecepatanY - kec0Y) /
((double)(CLOCKS_PER_SEC * 1) / CLOCKS_PER_SEC);

x_awal = x_akhir;
kec0X = kecepatanX;
y_awal = y_akhir;
kec0Y = kecepatanY;
double focal_length2 =
1969.93290799997;

double dot_pitch = 0.300858163;
double Xreal2 = (x_awal)-(0.5 * 640);

```

```

double Yreal2 = (0.785 * 480) -
(y_awal);
double tangenX = (Xreal2 /
focal_length2);
double tangenY = (Yreal2 /
focal_length2);
double sudutX = Math::Atan(tangenX) *
180/pi;
double sudutY = Math::Atan(tangenY) *
180/pi;
std::cout << "sudut X (rotasi) = " <<
sudutX << " derajat\n";
std::cout << "sudut Y (elevasi) = " <<
sudutY << " derajat\n";
std::cout << "Kecepatan X = " <<
kecepatanX << " pixel per second\n";
std::cout << "Percepatan X = " <<
percepatanX << " pixel per second kuadrat\n";
std::cout << "Kecepatan Y = " <<
kecepatanY << " pixel per second\n";
std::cout << "Percepatan Y = " <<
percepatanY << " pixel per second kuadrat\n" << endl << endl;

std::stringstream ss;

ss << "X,Y=" << points[1][0];
string disp1 = ss.str();
putText(image, disp1, Point
(points[1][0]), FONT_HERSHEY_PLAIN, 0.85, CV_RGB(255,
255, 75), 1.5);

std::stringstream ss1;
ss1 << "SudutX=" << sudutX;
string disp2 = ss1.str();

```

```

        putText(image, disp2, Point(10,420),
FONT_HERSHEY_PLAIN, 0.85, CV_RGB(225, 75, 255), 1.5);

        ss1 << " SudutY=" << sudutY;
        string disp3 = ss1.str();
        putText(image, disp3, Point(10, 420),
FONT_HERSHEY_PLAIN, 0.85, CV_RGB(225, 75, 255), 1.5);

        std::stringstream ss2;
        ss2 << "Vx=" << kecepatanX;
        string disp4 = ss2.str();
        putText(image, disp4, Point(10, 435),
FONT_HERSHEY_PLAIN, 0.85, CV_RGB(75, 255, 225), 1.5);

        ss2 << " Vy=" << kecepatanY;
        string disp5 = ss2.str();
        putText(image, disp5, Point(10, 435),
FONT_HERSHEY_PLAIN, 0.85, CV_RGB(75, 255, 255), 1.5);

        std::stringstream ss3;
        ss3 << "ax=" << percepatanX;
        string disp6 = ss3.str();
        putText(image, disp6, Point(10, 450),
FONT_HERSHEY_PLAIN, 0.85, CV_RGB(255, 255, 255), 1.5);

        ss3 << " ay=" << percepatanY;
        string disp7 = ss3.str();
        putText(image, disp7, Point(10, 450),
FONT_HERSHEY_PLAIN, 0.85, CV_RGB(255, 255, 255), 1.5);

    }
    if (addRemovePt && points[1].size() <
(size_t)MAX_COUNT)
    {
        vector<Point2f> tmp;

```

```

        tmp.push_back(point);
        cornerSubPix(gray, tmp, winSize, Size(-
1, -1), termcrit);
        points[1].push_back(tmp[0]);
        addRemovePt = false;

    }
    needToInit = false;
    imshow("Deteksi dan Lacak Objek", image);

    char c = (char)waitKey(10);
    if (c == 27)
        break;
    switch (c)

    {
        case 'c':
            points[0].clear();
            points[1].clear();
            break;
        }

    std::swap(points[1], points[0]);
    cv::swap(prevGray, gray);
}
return 0;
;}

```

Lampiran 2. Tabel Verifikasi Objek

NO	Detik ke- (s)	f	KOORDINAT				VERIFIKASI					
			X1	X2	Y1	Y2	VX	VY	AX	AY	tetaX	tetaY
1	(0 - 1)	1,97E+03	144,51	144,62	395,628	395,636	0,151	0,008	0,149	0,0078	-5,09E+00	-5,38E-01
2	(1 - 2)	1,97E+03	157,177	159,644	395,298	395,18	2,467	0,118	2,123	0,101	-4,65E+00	-5,35E-01
3	(2 - 3)	1,97E+03	284,513	286,586	395,731	395,707	2,073	0,024	2,073	0,018	-9,72E-01	-5,50E-01
4	(3 - 4)	1,97E+03	361,946	364,752	395,857	395,858	2,806	0,001	2,412	0,013	1,30E+00	-5,54E-01
5	(4 - 5)	1,97E+03	447,34	448,074	394,208	390,341	0,734	3,867	0,633	3,267	3,72E+00	-3,94E-01
6	(5 - 6)	1,97E+03	459,212	459,921	301,643	296,344	0,709	5,299	0,512	4,376	4,06E+00	2,34E+00
7	(6 - 7)	1,97E+03	452,691	452,184	165,894	162,604	0,507	3,29	0,357	2,877	3,84E+00	6,21E+00
8	(7 - 8)	1,97E+03	448,405	448,362	142,505	141,396	0,043	1,109	0,103	0,897	3,73E+00	6,81E+00
9	(8 - 9)	1,97E+03	437,04	436,727	78,753	77,54	0,313	1,213	0,213	0,889	3,39E+00	8,64E+00
10	(0 - 1)	1,97E+03	407,31	403,472	52,124	51,774	3,838	0,35	3,576	0,41	2,43E+00	9,37E+00

NO	Detik ke- (s)	f	KOORDINAT				PROGRAM					
			X1	X2	Y1	Y2	VX	VY	AX	AY	tetaX	tetaY
1	(0 - 1)	1,97E+03	144,51	144,62	395,628	395,636	0,153	0,0079	0,145	0,008	-5,107	-0,529
2	(1 - 2)	1,97E+03	157,177	159,644	395,298	395,18	2,46	0,121	2,125	0,103	-4,674	-0,528
3	(2 - 3)	1,97E+03	284,513	286,586	395,731	395,707	2,061	0,022	2,125	0,017	-0,989	-0,528
4	(3 - 4)	1,97E+03	361,946	364,752	395,857	395,858	2,799	0,001	2,401	0,012	1,28	-0,578
5	(4 - 5)	1,97E+03	447,34	448,074	394,208	390,341	0,802	3,901	0,621	3,207	3,719	-0,384
6	(5 - 6)	1,97E+03	459,212	459,921	301,643	296,344	0,721	5,314	0,505	4,345	4,038	2,349
7	(6 - 7)	1,97E+03	452,691	452,184	165,894	162,604	0,503	3,192	0,346	2,977	3,835	6,225
8	(7 - 8)	1,97E+03	448,405	448,362	142,505	141,396	0,041	1,101	0,099	0,901	3,719	6,829
9	(8 - 9)	1,97E+03	437,04	436,727	78,753	77,54	0,31	1,201	0,222	0,887	3,371	8,657
10	(0 - 1)	1,97E+03	407,31	403,472	52,124	51,774	4,123	0,372	3,582	0,417	2,413	9,393

NO	EROR					
	VX	VY	AX	AY	tetaX	tetaY
1	1,32%	1,25%	2,68%	2,56%	0,38%	1,65%
2	0,28%	2,54%	0,09%	1,98%	0,44%	1,23%
3	0,58%	8,33%	2,51%	5,56%	1,77%	3,98%
4	0,25%	0,00%	0,46%	7,69%	1,64%	4,28%
5	9,26%	0,88%	1,90%	1,84%	0,02%	2,50%
6	1,69%	0,28%	1,37%	0,71%	0,61%	0,44%
7	0,79%	2,98%	3,08%	3,48%	0,10%	0,31%
8	4,65%	0,72%	3,88%	0,45%	0,25%	0,21%
9	0,96%	0,99%	4,23%	0,22%	0,59%	0,22%
10	7,43%	6,29%	0,17%	1,71%	0,55%	0,26%
Mean Error	2,72%	2,43%	2,04%	2,62%	0,64%	1,51%

BIODATA PENULIS



Penulis dilahirkan di Jakarta, 24 November 1995, merupakan anak kelima dari 5 bersaudara. Penulis telah menempuh pendidikan tingkat dasar di Sekolah Dasar Negeri 15 Rawabunga Pagi. Sementara untuk pendidikan menengah, penulis menimba ilmu di SMP Negeri 109 Jakarta dan SMA Negeri 61 Jakarta. Kemudian setelah mengikuti Seleksi Nasional Masuk PTN, penulis diterima di Departemen Teknik Mesin FTI-ITS pada tahun 2013 dan terdaftar dengan NRP 02111340000053.

Di Departemen Teknik Mesin ini, penulis mengambil Bidang Studi Manufaktur. Penulis sempat aktif di beberapa organisasi tingkat Departemen yaitu DIMENSI Mesin ITS tingkat Institut yaitu Badan Eksekutif Mahasiswa ITS dan WE&T ITS. Selain itu, penulis juga aktif sebagai Asisten dan Grader Praktikum Pengukuran Teknik.

[Halaman ini sengaja dikosongkan]